

Omnilingual ASR: Open-Source Multilingual Speech Recognition for 1600+ Languages

Omnilingual ASR team, Gil Keren[†], Artyom Kozhevnikov[†], Yen Meng[†], Christophe Ropers[†], Matthew Setzler[†], Skyler Wang^{†,1}, Ife Adebare², Michael Auli^{*}, Can Balioglu, Kevin Chan, Chierh Cheng, Joe Chuang, Caley Droof, Mark Duppenthaler^{*}, Paul-Ambroise Duquenne, Alexander Erben, Cynthia Gao, Gabriel Mejia Gonzalez, Kehan Lyu, Sagar Miglani, Vineel Prapat^{*}, Kaushik Ram Sadagopan^{*}, Safiyyah Saleem, Arina Turkatenco, Albert Ventayol-Boada, Zheng-Xin Yong^{*,3}, Yu-An Chung[‡], Jean Maillard[‡], Rashel Moritz[‡], Alexandre Mourachko[‡], Mary Williamson[‡], Shireen Yates[‡]

FAIR at Meta, ¹Department of Sociology, McGill University, ²Department of Modern Languages and Cultural Studies, University of Alberta, ³Department of Computer Science, Brown University, ^{*}Work conducted while at FAIR at Meta

[†]Core contributors, alphabetical order, [‡]Technical leadership and project management, alphabetical order

While automatic speech recognition (ASR) systems have made remarkable progress in many high-resource languages, most of the world’s 7,000+ languages remain unsupported, with thousands of long-tail languages effectively left behind. Expanding ASR coverage has long been regarded as prohibitively expensive and of limited benchmark value, further hampered by architectures that restrict language coverage to a fixed set that make extension inaccessible to most communities—all while entangled with ethical concerns when pursued without community collaboration. To transcend these limitations, this article introduces Omnilingual ASR, the first large-scale ASR system designed for extensibility. More specifically, Omnilingual ASR enables communities to introduce unserved languages with only a handful of their own data samples. On the modeling side, Omnilingual ASR scales self-supervised pre-training to 7B parameters to learn robust speech representations and introduces an encoder–decoder architecture designed for zero-shot generalization, leveraging a large language model-inspired decoder to effectively exploit these representations. This capability is grounded in a massive and diverse training corpus; by combining breadth of coverage with linguistic variety, the model learns representations robust enough to adapt to previously unseen languages. Incorporating public resources with community-sourced recordings gathered through compensated local partnerships, Omnilingual ASR expands coverage to more than 1,600 languages, the largest such effort to date—including over 500 never before served by any ASR system. Automatic evaluations show substantial gains over prior systems, especially in extreme low-resource conditions, and strong generalization to languages never encountered during training. Crucially, Omnilingual ASR is released as a family of models ranging from compact 300M variants for low-power devices to large 7B models for maximum accuracy. Throughout the paper, we reflect on the ethical considerations shaping this design and conclude by discussing its broader societal impact. In particular, we highlight how open-sourcing models and tools can lower barriers for researchers and communities alike, inviting new forms of participation without requiring onerous expertise or heavy compute. All open-source artifacts from this effort are available at <https://github.com/facebookresearch/omnilingual-asr>.

Date: November 10, 2025

Correspondence: Yu-An Chung at andyuan@meta.com, Jean Maillard at jeanm@meta.com

Code: <https://github.com/facebookresearch/omnilingual-asr>

Blogpost: <https://ai.meta.com/blog/omnilingual-asr-advancing-automatic-speech-recognition> 

Contents

1	Introduction	4
2	Speech Recognition for Long-Tail Languages	5
2.1	A Brief Overview of ASR	5
2.2	Overcoming challenges to Long-Tail ASR	6
3	Data and Language Coverage	7
3.1	Referring to Languages	7
3.2	Defining Language Coverage	8
3.3	Dataset creation	8
3.3.1	Existing ASR Data	8
3.3.2	Partner-Created ASR Data	9
3.3.3	Commissioned ASR Data: The Omnilingual ASR Corpus	9
3.3.4	Pre-training data	13
3.4	ASR Data Preparation and Cleaning	13
3.5	Final Datasets	14
4	Omnilingual ASR Models	15
4.1	Massively Cross-Lingual Self-Supervised Representations	15
4.1.1	Self-supervised Pre-training with wav2vec 2.0	15
4.1.2	Scaling Speech SSL Beyond 2B	16
4.2	Automatic Speech Recognition	17
4.3	Zero-Shot Speech Recognition for Unseen Languages	17
4.4	Selection of Context Examples for Zero-Shot ASR	19
4.5	Conditioning on Language Codes	19
5	Model Training and Evaluation	19
5.1	ASR Training Setup	20
5.2	Comparison to Other Work	20
5.2.1	Omnilingual ASR vs. Whisper	20
5.2.2	Omnilingual ASR vs. USM	22
5.2.3	Omnilingual ASR vs. MMS	22
5.3	Evaluation on 1600+ languages	23
5.3.1	Evaluation based on Resource Buckets	23
5.3.2	Evaluation based on Language Groupings	24
5.4	Accuracy of Zero-Shot Models on Unseen Languages	25
5.5	Constructing Context Examples for Zero-Shot ASR	26
5.6	Applications to Speech-to-Text Translation	27
5.6.1	S2TT Experimental Setting	28
5.6.2	S2TT Results and Discussion	28
5.7	Impact of Datamix	28
5.7.1	Upsampling Low-Resource Languages	29
5.7.2	Generalizing to Unseen Audio Distributions	29
5.7.3	Model Robustness to Background Noise	31
5.7.4	Omnilingual + OMSF ASR Holdout Ablation	32
5.7.5	Fine-tuning for Individual Low-Resource Languages	33
5.8	Impact of Conditioning on Language Codes	34
5.9	Comparison of OmniASR-W2V Models to Existing SSL Speech Encoders	36
6	Societal Impact and Conclusion	37
A	Omnilingual ASR Language Coverage	45

B	WER Filtering	49
C	Prompts and Guidelines for Commissioned Data Collection	49
C.1	Recording guidelines	49
C.2	Transcription guidelines	50
D	Quality Assurance (QA) Guidelines	51
D.1	Speech recording error taxonomy	51
D.2	Transcript error taxonomy	51

1 Introduction

Automatic speech recognition (ASR) has made extraordinary strides in recent years, with state-of-the-art systems approaching human-level accuracy in many high-resource languages (Radford et al., 2023; Pratap et al., 2024; Zhang et al., 2023). Yet beyond this small set lies the long tail of linguistic diversity—thousands of languages, most with little to no ASR support (Bartelds et al., 2023). Extending speech technology to this long tail is widely acknowledged as valuable, but in practice, it is rarely pursued at scale (Yadav and Sitaram, 2022).

Researchers often shy away from long-tail ASR for a mix of practical and ethical reasons. From a practical standpoint, expanding coverage to low-resource languages can be expensive, requiring substantial engineering and data collection infrastructure for comparatively small amounts of training data (Hussen et al., 2025). Moreover, the returns are often seen as modest: a large investment may yield little improvement in benchmark performance, and the work may be perceived as less “impactful” than progress in dominant languages or novel model architectures. From an ethical standpoint, there is a concern that building technology for under-resourced communities without careful calibration risks disempowering those very communities, raising questions about language ownership and sovereignty (Choi and Choi, 2025; Reitmaier et al., 2022).

While these concerns are real and deserve sustained attention, the prevailing hesitancy has important drawbacks. First, the assumption that long-tail ASR impact is minimal ignores the fact that for many communities, even modest ASR capabilities can be transformative—making oral archives searchable, enabling voice-driven interfaces in one’s own language, and contributing to the revitalization of endangered languages (Mainzinger and Levow, 2024). Second, the notion that such work lacks scientific value overlooks the unique technical challenges of the long tail: extreme data scarcity, orthographic variability, and phonetic diversity that can push the limits of model design and learning architectures (Imam et al., 2025). Finally, the fear of ethical missteps should be addressed not by withdrawal, but by building frameworks for social-centered and community-driven collaboration (Cooper et al., 2024; Reitmaier et al., 2022; Wang et al., 2024b)—supported by transparent open-sourcing of models and evaluation tools to enable local adaptation and control (NLLB Team, 2024; SEAMLESS Communication Team, 2025). Just as importantly, new architectures and design choices can be developed with community agency in mind, shifting innovation away from one-size-fits-all models toward systems that are extensible and co-shaped with the speakers who use them.

With that in mind, this paper introduces **Omnilingual ASR**, a state-of-the-art multilingual speech recognition system that redefines how language coverage in this domain is approached. Beyond expanding to over 1,600 languages, the largest such effort to date and including more than 500 that have never been supported by any ASR system (see Section A for the full list), Omnilingual ASR also shifts the paradigm for how *new* languages can be brought into the fold. In most existing systems, languages not included at release can only be added through expert-driven fine-tuning—a path inaccessible to most communities. Omnilingual ASR instead introduces the first large-scale ASR framework capable of extending to entirely new languages with just a few in-context examples. This capability is enabled by an encoder-decoder architecture designed for zero-shot generalization, scaling self-supervised pre-training to 7B parameters to extract speech representations, then exploiting them with a large language model (LLM)-inspired decoder. In practice, this means that a speaker of an unsupported language can provide only a handful of paired audio-text samples and obtain reasonable transcription quality—without training data at scale, out-of-reach expertise, or access to high-end compute. While zero-shot performance cannot yet match that of fully trained systems, it offers a far more scalable path to bringing new languages into digital reach.

Omnilingual ASR also advances the state of multilingual ASR along more familiar dimensions. Its training corpus is one of the largest ever assembled for ASR in both volume and linguistic diversity, integrating publicly available datasets with community-sourced speech recordings collected through commissioned partnerships. To reach languages with little or no digital presence, we worked with local organizations who recruited and compensated native speakers, often in remote or under-documented regions. Evaluations across diverse benchmarks show consistent quality improvements over prior systems, particularly in low-resource settings, and demonstrate strong generalization to languages never encountered during training. To promote adoption in both research and deployment contexts, Omnilingual ASR is released not as a single model but as a family—ranging from large 7B-parameter variants to compact 300M-parameter versions that can run on

low-power devices “in the wild.”

By enabling the ability to support languages beyond the predefined set, at the initiative of speakers themselves, Omnilingual ASR changes the terms of long-tail ASR. No model can ever anticipate and include all of the world’s languages in advance, but Omnilingual ASR makes it possible for communities to extend recognition with their own data—without large-scale training or specialized expertise. This reframes ASR coverage not as a static inventory but as an extensible framework, opening space for community-driven adaptation and agency. Throughout the paper, we reflect on the ethical considerations guiding this approach, and we conclude by discussing the broader societal impact of enabling speech technology for the world’s long-tail languages.

To spur further research and enable community-driven expansion, we open-source the following at <https://github.com/facebookresearch/omnilingual-asr>:

- a suite of self-supervised (SSL) pre-trained speech models that come in 300M, 1B, 3B, and 7B parameters, all of which cover 1600+ languages suitable for fine-tuning for a wide range of downstream speech tasks and varying computational conditions;
- a suite of supervised connectionist temporal classification (CTC) based ASR models fine-tuned from the SSL checkpoints suitable for basic ASR applications with strong performance;
- a suite of supervised LLM-based ASR models for state-of-the-art ASR performance;
- a zero-shot LLM-based ASR model that transcribes utterances of unseen languages using only a few examples provided at inference time;
- a massively multilingual ASR dataset covering over 300 languages, with an average of 10 hours of transcribed speech per language; for many languages, this represents the first ASR corpus ever built.

2 Speech Recognition for Long-Tail Languages

2.1 A Brief Overview of ASR

ASR has long been imagined as a cornerstone of human–computer interaction, with early systems in the mid-20th century only able to recognize digits or a few carefully scripted words (Davis et al., 1952). Over the decades, research steadily expanded the scope of what ASR could do, from isolated command-and-control vocabularies to continuous recognition of natural speech (Young, 1996). A critical driver of this progress was the availability of benchmark datasets that allowed researchers to measure advances and refine algorithms in widely spoken languages like English (Garofolo et al., 1993). By the 2010s, with the rise of deep learning, ASR reached a turning point: feedforward deep neural networks (DNNs) and later recurrent neural networks (RNNs) drastically improved acoustic modeling, while sequence-to-sequence and attention-based architectures laid the foundation for fully end-to-end ASR systems (Chorowski et al., 2015; Graves and Jaitly, 2014). Large public corpora like LibriSpeech (Panayotov et al., 2015), derived from audiobooks, further accelerated progress by standardizing evaluation in English. Systems trained on large amounts of labeled data began approaching human-level accuracy for certain high-resource languages, and speech technology entered everyday applications from voice assistants to automated captioning (Radford et al., 2023).

The more recent wave of progress has been propelled by scaling—both in terms of training data and model architectures. Datasets such as MLS (Pratap et al., 2020), VoxPopuli (Wang et al., 2021), MSR (Li et al., 2024) and Granary (Koluguri et al., 2025) have substantially increased the amount of transcribed speech available for training, though these advances have been directed mostly at languages which were already high-resource. Efforts to include lower-resource languages have accelerated in recent years, with datasets such as BLOOM (Leong et al., 2022) covering 56 languages, Speech Wikimedia (Gómez et al., 2023) reaching 77, and YODAS (Li et al., 2023) spanning 140. Yet despite these expansions, the distribution of data remains heavily skewed, and only a handful of recordings exist for many of the most under-served languages. A broader coverage of nearly 700 languages is offered by CMU wilderness (Black, 2019), which was derived from publicly available Bible recordings and therefore lacks diversity in domain, reading style, and speakers. An analogous effort that is primarily restricted to the religious domain is the MMS dataset (Pratap et al., 2024), reproduced in its untranscribed part by Chen et al. (2024), representing the largest coverage to date with

over 4,000 languages. Of particular note are projects such as VAANI (Team, 2025), which is dedicated to the collection of natural speech in over 100 languages from the Indian subcontinent, and African Next Voices (Marivate et al., 2025; KenCorpus Consortium, 2025; Digital Umuganda, 2025a,c,e,d,b), which focuses on providing large, high-quality and culturally rich datasets for African languages. Common Voice (Ardila et al., 2020)—maintained by the Mozilla Foundation and curated by a large network of volunteers—currently spans approximately 130 languages and stands out as the most extensive and widely utilized datasets.

Advancements made in self-supervised learning have further reshaped the field. More specifically, models like wav2vec 2.0 (Baeovski et al., 2020) demonstrate how massive amounts of unlabeled audio could be leveraged to learn powerful speech representations, drastically reducing the need for labeled data. This paradigm enabled breakthroughs such as the Universal Speech Model by Zhang et al. (2023), pre-trained on 12 million hours of unlabelled speech spanning over 300 languages and fine-tuned on a smaller labeled dataset, and the MMS model of Pratap et al. (2024), which extended coverage beyond 1,100 languages through large-scale pre-training. Self-supervision can also improve the language modeling or text generation component of ASR systems, including in multilingual settings, as demonstrated by works by Babu et al. (2021), Bapna et al. (2022), and Pratap et al. (2024). Moreover, architectural innovations can allow models to transcribe languages unseen during training. For instance, Li et al. (2022) propose an approach based on mapping the output of an 8-language multilingual model to language-specific phonemes, a method extensible to any unseen languages which have n-gram statistics, though limited by the reliability of phoneme mappings for low-resource languages. Building on this, Zhao et al. (2025) remove the intermediate phone representations and instead adopt a romanization-based encoding, achieving lower error rates. Although recent advances in language adaptation and zero-shot capabilities of large language models show promise (Yong et al., 2023), these gains have so far accrued mainly to high-resource languages (Ahuja et al., 2023; Bang et al., 2023; Asai et al., 2024; Ochieng et al., 2025).

2.2 Overcoming challenges to Long-Tail ASR

From above, we see that despite recent achievements in the field of ASR, the benefits remain concentrated in a relatively small subset of high-resource languages, leaving the vast majority of the world’s linguistic diversity unsupported. Understanding why such an important problem is rarely undertaken at scale requires unpacking the practical, scientific, architectural, and political barriers that have kept many long-tail languages on the margins of ASR development. Below, we outline some of these hurdles.

Practical barriers. Collecting training data for low-resource languages is resource-intensive. Unlike high-resource languages, which have vast amounts of texts and transcribed speech available, many long-tail languages require costly, ground-up data creation (Abraham et al., 2020; Besacier et al., 2014). This often involves recruiting native speakers, designing orthographic conventions, and collecting high-quality audio in settings where infrastructure may be limited. The effort is large, yet the resulting datasets are comparatively small, making them less attractive for institutions prioritizing efficiency or scale (Blasi et al., 2021).

Scientific disincentives. In the research community, progress is typically measured by benchmarks and leaderboard gains. Improving ASR for a long-tail language rarely moves the needle on widely used benchmarks, and therefore can be perceived as less “impactful” or publishable (Mainzinger and Levow, 2024). The challenges are also technically demanding: extreme data scarcity, phonetic diversity, and variable orthographies stretch existing architectures beyond their tested limits (Adda et al., 2016; Joshi et al., 2020). These are precisely the kinds of challenges that could advance the science of ASR, but in practice they often push researchers toward safer ground.

Architectural limitations. Existing ASR systems generally treat language coverage as fixed at release. If a language is not included in training, extending support typically requires expert-driven fine-tuning with large compute resources and specialized expertise—an approach inaccessible to most communities (Imam et al., 2025). This lack of extensibility effectively prevents many groups from bringing their languages into digital spaces, slowing progress toward inclusive ASR.

Ethical and political complexities. Long-tail languages are deeply entangled with questions of identity, ownership, and sovereignty. Building ASR systems without community involvement risks creating extractive dynamics (Bird, 2024), where outside institutions “take” language data without returning meaningful benefits to speakers.

Concerns about appropriation or misuse have led some researchers to avoid long-tail ASR altogether, fearing that well-intentioned efforts might inadvertently disempower the very communities they aim to support (Choi and Choi, 2025; Cooper et al., 2024).

While these practical and ethical concerns explain the historical neglect of long-tail languages, leaving them unsupported is far from a neutral choice. The lack of ASR capacity has tangible consequences for the communities situated at the margins (Joshi et al., 2020). Many of these languages are primarily oral, with few standardized orthographies or written resources. Without ASR, oral archives—from folktales to political speeches—remain locked in raw audio, inaccessible to researchers, educators, or even community members seeking to preserve and circulate their own heritage. In more everyday terms, the absence of speech technology excludes entire populations from tools that dominant-language speakers take for granted: dictation, search, subtitling, or voice-based accessibility services (Mainzinger and Levow, 2024). This exclusion is not simply technical; it reinforces digital hierarchies in which only speakers of globally dominant languages can fully participate in an increasingly voice-driven digital ecosystem (SEAMLESS Communication Team, 2025). For minority communities, the effects can be even more acute, as the lack of technological affordances accelerates language shift: younger speakers may turn toward dominant languages that provide digital tools, leaving their heritage languages further marginalized (Kornai, 2013).

This current effort hopes to transcend these barriers by recognizing that inaction perpetuates inequality. Not building ASR for long-tail languages is itself a decision—one that deepens digital divides and risks silencing already vulnerable voices. To counter this, our approach prioritizes community partnerships, ensuring that the extension of ASR coverage is developed collaboratively with local actors. By working directly with communities, compensating native speakers for speech data, and enabling local adaptation through open-source release, Omnilingual ASR aims not only to expand technical coverage but to lay the groundwork for more inclusive, community-driven participation in the speech technology ecosystem.

3 Data and Language Coverage

Building a system that can recognize and transcribe speech across more than 1,600 languages first required the largest and most diverse ASR training corpus assembled to date. Achieving this breadth meant integrating resources from multiple domains: existing public datasets, internal collections developed for prior multilingual ASR systems, and crucially, community-sourced speech recordings that extend coverage into languages with little or no prior digital footprint. In this section, we provide additional information about language coverage and break down the training corpus creation process.

3.1 Referring to Languages

In the absence of a strict scientific definition of what constitutes a *language*, we adopted a practical convention: treating as candidate languages those linguistic entities—*languoids*, following Good and Hendryx-Parker (2006)—that have been assigned their own ISO 639-3 codes.

We acknowledge that language classification in general, and the attribution of ISO 639-3 codes in particular, is a complex process, subject to limitations and disagreements, and not always aligned with how native speakers themselves conceptualize their languages. To allow for greater granularity when warranted, ISO 639-3 codes were complemented with Glottolog languoid codes (Hammarström et al., 2024). For example, we preserved the distinction between the Vallader and Sutsilvan varieties of Romansh, following the practice of the Mozilla Common Voice community, by using the Glottocodes `lowe1386` and `suts1235`. In the rare cases where Glottolog’s classification is known but actively disputed by the speaker communities we worked with, we supplemented ISO 639-3 codes with community-supported languoid names; for instance, by adopting the IANA language variant subtags `gherd` and `valbadia` for Ladin.

Due to the written component of the ASR task, careful attention was also paid to languages with multiple writing systems. Accordingly, all languages supported by our model are associated with one or more ISO 15924 script codes. Take Mandarin, for example, we use `cmn_Hant` to denote Mandarin Chinese in the traditional script and `cmn_Hans` for the same language in the simplified script. Where additional variants are needed,

we extend this system; for example, `roh_Latn_suts1235` identifies the Sutsilvan Romansh languoid written in the Latin script.

3.2 Defining Language Coverage

For ASR applications, at least some of the training data must consist of speech recordings paired with transcripts. The first steps in defining language coverage are therefore to ensure, first, that the language candidates are spoken, and second, that they have an established writing system. Both points warrant brief elaboration.

First, the ISO 639-3 inventory (with more than 7,000 codes) includes roughly 150 signed languages. Because these are not spoken, they cannot be directly included in ASR applications. Second, the availability and classification of writing systems is far from straightforward. It is not a simple dichotomy between written and unwritten languages. Some languages consistently employ a single writing system, while others have used multiple systems either historically or concurrently. In certain cases, these practices are well documented; in others, information is incomplete or missing. For instance, ScriptSource¹ reports 2,586 languages with insufficient information on their scripts. This does not imply that the languages in question are unwritten, but it does highlight the challenges of securing textual data for them.

Our approach was to include only languages with at least one established writing system. By “established,” we mean a form of writing that is in frequent use, intelligible to the speaker community, and ideally described in formal resources such as dictionaries or grammars. This excludes transcriptions in the International Phonetic Alphabet² or idiosyncratic note-taking systems, which do not constitute stable or widely recognized orthographies.

Beyond the above considerations, additional steps were taken to define the scope of our language coverage while avoiding overlapping or redundant inclusion. Overlap can occur through macrolanguage codes or through duplication with already available data. Macrolanguage codes are a known feature of ISO 639-3. The standard defines 63 such codes, which may be used either to group related varieties or as a placeholder where more specific identification is unavailable. However, many macrolanguage codes are overly broad and often redundant. For example, the macrolanguage code `msa` for the Malay group of languages encompasses 36 other ISO 639-3 codes, including Indonesian and Minangkabau. To minimize ambiguity, such macrolanguage codes were excluded wherever possible. Lastly, we also deprioritized languages already covered in prior ASR work, such as Pratap et al. (2024), on which Omnilingual ASR builds. Finally, constructed languages and languages classified by UNESCO as extinct were also deprioritized, as neither provide a viable basis for ASR applications.

3.3 Dataset creation

Building Omnilingual ASR involved compiling the largest linguistically diverse speech dataset ever created. In this section we detail the extensive efforts undertaken to assemble existing resources and develop new ones through partnerships and commissioning.

3.3.1 Existing ASR Data

We assembled training data from a large number of existing open-source datasets: ALFFA (Abate et al., 2005; Gelas et al., 2012; Gauthier et al., 2016), LibriSpeech ASR (Panayotov et al., 2015), the South African language data of van Niekerk et al. (2017), ASR and TTS data by Kjartansson et al. (2018), Sodimana et al. (2018) and He et al. (2020), CSS10 (Park and Mulc, 2019), FOSD (Tran, 2020), Zeroth Korean dataset,³ Burmese Speech Corpus (Oo et al., 2020), Common Voice v22 (Ardila et al., 2020), VoxPopuli (Wang et al., 2021), VoxLingua-107 (Valk and Alumäe, 2021), RuLS,⁴ the Kokoro Speech Dataset,⁵ MLS (Pratap et al., 2020), Samrómur (Mollberg et al., 2020), the Kazakh Speech Corpus (Khassanov et al., 2021), iMaSC (Gopinath

¹<https://scriptsource.org/entry/wekytdkcc> (retrieved 2025-08-19)

²International Phonetic Alphabet

³<https://github.com/goodatlas/zeroth>

⁴<https://www.openslr.org/96/>

⁵<https://github.com/kaiidams/Kokoro-Speech-Dataset>

et al., 2022), ParlaSpeech-HR (Ljubešić et al., 2022), NPSC (Solberg and Ortiz, 2022), FLEURS (Conneau et al., 2023) and NaijaVoices (Emezue et al., 2025).

We supplemented these sources with additional ASR data, coming from an internal dataset of publicly available speech recordings paired with transcriptions, and a number of commercially-available licensed datasets including the 17 language packs from the IARPA Babel program (Gales et al., 2014).

Finally, we integrated these resources with datasets shared from partners taking part in our Language Technology Partner Program, an effort intended to offer opportunities for interested members of the public to contribute to AI language technologies, with a particular focus on under-served languages. Participating members were able to access technical workshops led by our research team, learning how to leverage open-source models to build language technologies for their languages.

3.3.2 Partner-Created ASR Data

To support the development of speaker-centric ASR datasets, we provided funding and additional resources for several collaborative initiatives that placed native speakers and local communities at the center of the process, ensuring that the data collected was truly reflective of their linguistic and cultural input.

One such key effort is the African Next Voices project, a consortium led by Maseno University in Kenya, University of Pretoria in South Africa and Data Science Nigeria, aiming to bridge the technological divide in speech technologies for African languages and to promote equitable AI development across the continent. This project—which is supported by the Gates Foundation—ultimately aims to provide tens of thousands of hours of ASR data for up to twenty of the continent’s most spoken languages. The significant progress from this ongoing initiative is well documented in numerous scientific papers and open-source artifacts (Marivate et al., 2025; KenCorpus Consortium, 2025; Digital Umuganda, 2025a,c,e,d,b,f,g).

Additionally, we provided support to the Open Multilingual Speech Fund by Mozilla Foundation’s Common Voice (Ardila et al., 2020). This empowered over 170 new language communities to join the project. This support for community-centered open data work has enabled the number of communities participating in Common Voice to more than double. It brings the Common Voice corpus to well over 300 languages, helping to enrich linguistic diversity in technology for everyone.

Finally, we supported the Lanfrica/Naijavoices initiative,⁶ which resulted in the creation of new datasets for 11 African languages (Bainouk-Gunyaamolo, Balanta-Kentohe, Bube, Fang, Igala, Central Kanuri, Karon, Nupe-Nupe-Tako, Upper Guinea Crioulo, Serer and Urhobo) with a focus on high-quality, culturally representative, and demographically diverse content.

3.3.3 Commissioned ASR Data: The Omnilingual ASR Corpus

In addition to drawing on the aforementioned resources, we commissioned a tailored set of recordings and transcriptions to strengthen the corpus. This step ensured that the model would be trained on domain-diverse, high-quality spontaneous speech spanning a broad range of languages. By proactively filling gaps left by prior datasets, we aimed to create a resource that not only meets the immediate needs of this project but also enhances the model’s long-term adaptability. As we show in Sections 4.3 and 4.4, this diverse foundation is already demonstrating its value by facilitating cross-lingual transfer through zero-shot generalization. Below, we document the steps taking to develop the Omnilingual ASR Corpus, all of which is open-source and be made publicly available.

Prompt design. Our initial goal was to commission the collection of 10 hours of speech from 10 different native speakers (1 hour per speaker) in each for roughly 350–400 languages, paired with corresponding transcripts. To elicit naturally occurring language grounded in speakers’ experiences while avoiding personal information, we developed survey-style prompts such as *Is it better to have a few close friends or many casual acquaintances? Why?* Vendors were provided with a pool of more than 1,500 such prompts, ensuring sufficient material for one hour of naturally-occurring speech. The prompt set was made available in English and six pivot languages (French, Indonesian, Italian, Mandarin Chinese, Portuguese, and Spanish).

⁶<https://naijavoices.com/>



(a) Local participants contributing to corpus creation efforts in Pakistan.



(b) Local participants contributing to corpus creation efforts in Liberia.



(c) Example of the difficult travel conditions encountered during fieldwork.

Figure 1 Photographs documenting key moments from the global collection of speech data that produced the Omnilingual ASR Corpus.

Importantly, we deliberately over-supplied prompts—far more than any speaker would need for a single session. This decision served several purposes. First, no single set of questions can feel equally relevant worldwide; by offering breadth, we allowed participants to skip prompts they found uncomfortable or uninteresting. Second, the abundance of options let speakers guide the recordings toward topics they cared about, fostering engagement and spontaneity. In practice, many participants moved fluidly between prompts and their own digressions. For example, one speaker began with a lighthearted role-play prompt about imagining life as a bird and ended with a detailed reflection on the nesting habits of local bird species. This design ensured that our dataset was not only broad and balanced but also enriched with authentic, culturally grounded, and participant-driven speech.

Native speaker availability. In practice, it was not always possible to follow the initial collection plan exactly. First, suitable speakers could not be found in all languages within the specified time frame. In some cases, this meant that the 10-speaker target was not met, reducing the total amount of collected recordings and transcripts. In others, the shortfall was offset because available speakers recorded more than one hour each, allowing the 10-hour target to be met even without 10 distinct contributors. A further set of languages had speakers recruited but did not complete the full collection in time for inclusion in the training mix; nonetheless, we release those recordings and transcripts as part of the final open-source dataset. Finally, in a positive deviation from plan, vendors were able to document established writing systems for some languages not initially listed as candidates, and proceeded to collect speech recordings and transcripts for them as well. [Table 3](#) summarizes basic statistics on all training data, including the commissioned data collection to date (Omnilingual ASR Corpus).

Recordings. Participants were provided with prompts (or, in some cases, had prompts read aloud to them) and asked to respond. Prompts could be delivered either in participants’ native languages or in a second language in which they were proficient, but responses were to be given in their native languages, spoken naturally and at a normal pace—neither rushed nor artificially slow. When references to foreign terms were needed, participants were encouraged to pronounce them as they ordinarily would when speaking with fellow native speakers. Finally, participants were instructed to avoid sharing any personally identifiable information (PII), with a full list of items considered so provided in [Section C.1](#).

Transcripts. For the purpose of building ASR datasets, speech recordings must be paired with accurate transcriptions. We define accuracy here in two dimensions: first, transcriptions should be produced in an established writing system for each language (see [Section 3.2](#)); second, they must adequately reflect the characteristics of naturally-occurring spontaneous speech.

Unlike scripted or prepared speech, spontaneous speech exhibits disfluencies (repetitions, false starts, repairs, or incomplete sentences). These occur alongside non-verbal vocalizations such as fillers, laughter, breathing sounds, or coughs. To ensure faithful transcripts, such events must be annotated, along with occasional non-vocal sounds and background noise. For this purpose, participants were asked to use special tags—`<laugh>`, `<hesitation>`, `<unintelligible>`, and `<noise>`. Further details on tag usage are provided in the transcription guidelines (see [Section C.2](#)).

In addition to typical challenges that stem from the complexity of accurate spontaneous speech transcription in any language, more specific challenges also arise when attempting to transcribe low-resource languages, many of which are facing intergenerational disruption ([Fishman, 1991](#)). It is not uncommon for native speakers of disrupted languages to reside in more rural areas, where getting access to digital devices that produce and store machine-readable transcripts can be a challenge. Even when such devices are available, they may not support the relevant script or orthography. It might also happen that speakers who have native mastery of the spoken language do not feel as comfortable with its written form. For these reasons, transcripts were not always produced by the speakers themselves. In some cases, they were prepared by on-site typists; in others, handwritten notes were later digitized off-site. Each degree of separation from the original speaker introduced additional challenges to achieving transcription accuracy.

Quality assurance (QA). [Figure 2](#) shows the process by which the quality of the commissioned data was controlled. First, at the partial delivery stage, files were automatically screened for major quality flaws, such as corruption during transfer, unexpected duration, or excessive noise levels. A small number of files per language were also manually inspected by linguists, prioritizing those files that returned unexpected automated check results. After these initial rapid quality checks, feedback was communicated to vendors for easier root-cause identification and error correction. Then, at the final delivery stage, both speech and text data were uploaded to a specifically designed QA platform, and were inspected by trained QA technicians.

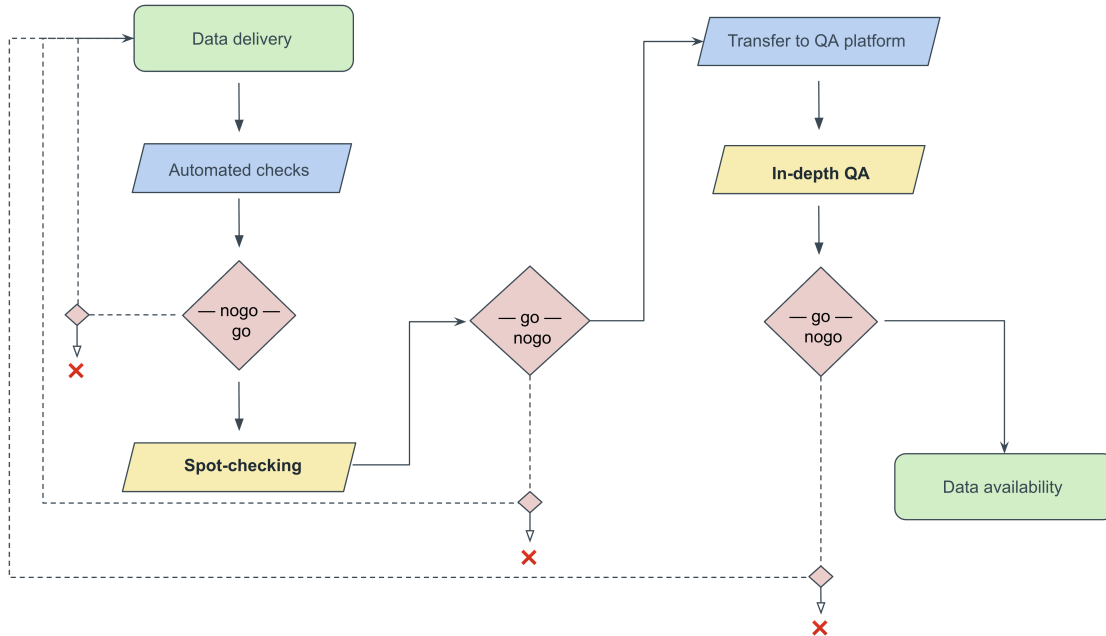


Figure 2 Commissioned data quality-assurance workflow.

The QA platform enabled technicians to access each speech recording alongside its corresponding transcript within a single interface, which also displayed the quality questionnaire they were required to complete. The

primary objectives of this task were to detect potential errors and classify them as either minor or critical. [Table 1](#) provides definitions for the most common error types in both categories, while a detailed description of the QA procedure and error taxonomy for speech recordings and transcripts is provided in [Section D](#).

Category	Critical example	Minor example
Human vocal noise	Second voice in the background Singing in the background	N/A (This error is always critical)
Cutoff	Speech is cut off at either end of the recording	N/A (This error is always critical)
Background noise	Rooster crowing Street noise, car honking Bird chirping Strong wind	Occasional mild coughing Occasional mild coughing Mild breathing sound

Table 1 Description of the error categories used for in-depth quality assurance (audio files)

Every language in the Omnilingual ASR Corpus went through at least the first step of human review (small-scale inspection), and 279 languages went through in-depth inspection. When rework was possible, quality issues were mitigated. In other cases, the portion of the data that did not meet quality requirements was excluded.

The QA process was instrumental in detecting and mitigating issues in data deliveries. Considering both minor and critical errors, the most frequent problems in audio files were long silences and background noise, while transcript files most often exhibited spelling inconsistencies and mismatches. Spelling inconsistencies are common in low-resource languages, where orthographies are not standardized in the same way as they are in high-resource languages. Mismatches between speech and transcripts, by contrast, are more serious but relatively straightforward to fix when identified early, as they usually reflect file misalignments rather than transcription errors per se.

Focusing on critical errors specifically, [Table 2](#) provides a more detailed breakdown of the six most prevalent categories. After long pauses, the most prevalent critical issues in speech recordings were cutoffs and human vocal noises. Cutoffs are likely the result of the recording equipment being mishandled, while vocal noises typically arose from audible human voices captured in the background.

Critical audio issues	Percentage of files	Critical transcript issues	Percentage of files
Pause / Silence	27.25%	Mismatch	51.18%
Cutoff	15.62%	Incomplete or summarized	21.97%
Human vocal noise	10.62%	Wrong writing system	10.51%
Background Noise	9.42%	Wrong tags	8.20%
Unnatural speech	9.05%	Numbers	1.97%
Low volume	5.31%	Inconsistent tagging	1.44%

Table 2 Most prevalent critical quality issues in speech and transcripts files

Validation. [Kreutzer et al. \(2022\)](#) show that a common quality issue in large, massively multilingual datasets stems from dataset mislabeling; i.e., the misattribution of language codes to some subsets of the data corpus. Such misattributions can be caused by several factors: for example, the use of both a private code and an attributed ISO code for the same language. Languages are often also known by different names in English and other languages, and even by different autonyms within their own native speaker groups. When the name of a language appears to be absent from the list of language names that correspond to ISO codes, it is tempting to create a private code without realizing that the language already has its ISO code under a slightly (or not so slightly) different name. Another type of code misattribution can come from a confusion between the code for a spoken language and the code for a sign language by a similar name (e.g., Hausa [hau] and Hausa Sign Language [hsl]).

To mitigate language code misattribution issues in the commissioned data, a validation project was set up whereby a small portion of the data collected by one vendor for a particular language was analyzed by a different vendor. The volume per language ranged between 1 to 5 audio files and up to 10 transcripts. For each sample audio and transcript file, proficient speakers of the target language were asked to determine whether the sample represented acceptable spoken or written forms of their language. Vendors were given additional guidance as to potential miscommunication due to the language naming discrepancies previously mentioned, as well as to discrepancies in the use of the terms *language* and *dialect*.

The language code validation process was applied to 206 languages, and allowed us to identify instances of misattributed language codes in 20 languages. These findings further underscore the significant challenges associated with collecting accurate data for Arabic and Fula languages in particular. The validation process also indirectly helped identify and correct a general language code attribution error for [zga]. For clarity, this language code validation step only constitutes additional due diligence on a very small portion of the datasets. The results of this process, whether negative or positive, should not lead to generalizations about entire datasets. Nevertheless, they provided additional insights into the quality of the commissioned data and into opportunities for improvement.

3.3.4 Pre-training data

As we will go into details in [Section 4.1](#), Omnilingual ASR is built on a massively multilingual speech encoder capable of producing high-performing cross-lingual speech representations. Training this encoder required a large-scale corpus of unlabeled speech. To construct it, we combined all the sources described in the preceding sections that were available when encoder training began. This phase predated the fine-tuning of the ASR models by several months, as well as the full delivery of our Omnilingual ASR Corpus and several partner-contributed ASR datasets. To further expand coverage, we supplemented these resources with a large-scale internal collection of unlabeled speech. The final pre-training dataset comprised 3.84M hours of speech across 1,239 languages, in addition to another 460K hours of speech for which no language identification was performed.

3.4 ASR Data Preparation and Cleaning

Concretely, we first split the text using the **sat-12l-sm** SAT model from [Frohmman et al. \(2024\)](#). By leveraging its splitting probability outputs, we ensured that text segments remained shorter than 200 characters. Annotators had often already inserted sentence boundaries, and SAT segmentation typically rediscovered this structure. However, for languages entirely out of SAT’s training domain and without sentence-level annotations, segmentation was instead driven by the maximum length constraint, without necessarily following sentence structure. Next, we applied a forced-alignment algorithm to obtain corresponding audio segments, following the procedure described in [Pratap et al. \(2024\)](#). If some audio segments remained too long ($> 50, s$), we reapplied the split-align operation with a reduced maximum text-segment length. Conversely, if audio segments were too short ($< 2, s$), they were merged with the nearest neighboring segment. Several iterations of split/merge ensured that final segments fell within the target range of $[2, s, 50, s]$. Finally, we note that no utterance-level segmentation was performed on existing public datasets such as FLEURS, MLS, or Babel.

After utterance-splitting, we applied WER-based filtering on the Omnilingual ASR Corpus to remove misaligned audio–text pairs. Such problematic examples were rare and typically arose either from erroneous reference transcripts or pathological edge cases in the segmentation/alignment pipeline. For curation, we used a 7B CTC model trained on a subset of available ASR data (excluding MLS, which does not contribute to lower-resource language coverage). We computed WERs for each utterance in the Omnilingual ASR Corpus datasets, then conducted qualitative analyses within each datasource to establish source-specific thresholds. Our philosophy was to apply minimal filtering, retaining as much data as possible while removing only clearly erroneous pairs. [Section B](#) provides the thresholds used as well as examples of filtered misalignments.

Finally, we constructed a character-based tokenizer by taking the union of all characters across the entire ASR dataset. This inventory was manually cleaned to remove obvious artifacts (e.g., punctuation, emojis) and extremely rare characters (occurring fewer than five times across the corpus) in order to limit vocabulary size. The resulting tokenizer contained 9,812 symbols. We then applied it to filter out degenerate transcripts containing $\geq 15\%$ unknown tokens.

3.5 Final Datasets

Once data preparation was complete, we combined all cleaned ASR datasets described in [Sections 3.3.1 to 3.3.3](#) into a unified corpus, which we refer to as ALLASR. Summary statistics of ALLASR are shown in [Table 3](#), and its overall distribution is illustrated in [Figure 3](#). Beyond expanding language coverage, consolidating diverse ASR corpora into a single dataset improved model robustness to varied audio conditions, as demonstrated in [Section 5.7.2](#).

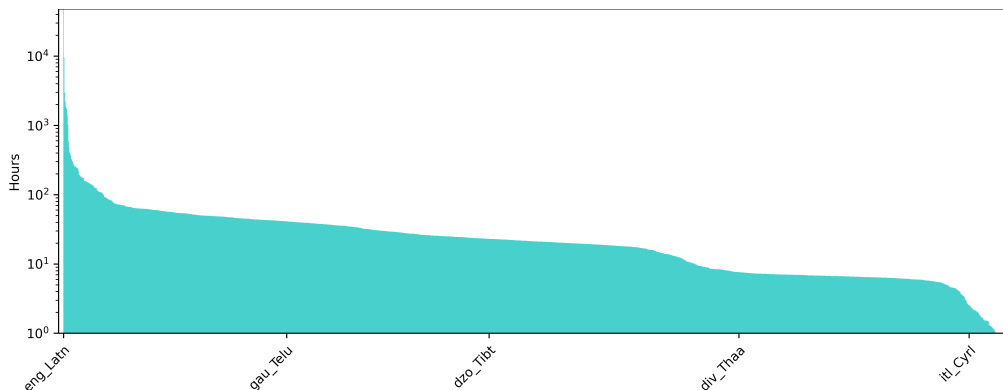


Figure 3 Statistics of the ALLASR labeled data (hours of speech recordings paired with transcription) used to pre-train Omnilingual ASR.

In parallel, the unlabeled speech data described in [Section 3.3.4](#) was consolidated into a single corpus for self-supervised pre-training. Long recordings were segmented into chunks no longer than 30,s to standardize training inputs. The overall distribution of this dataset is shown in [Figure 4](#).

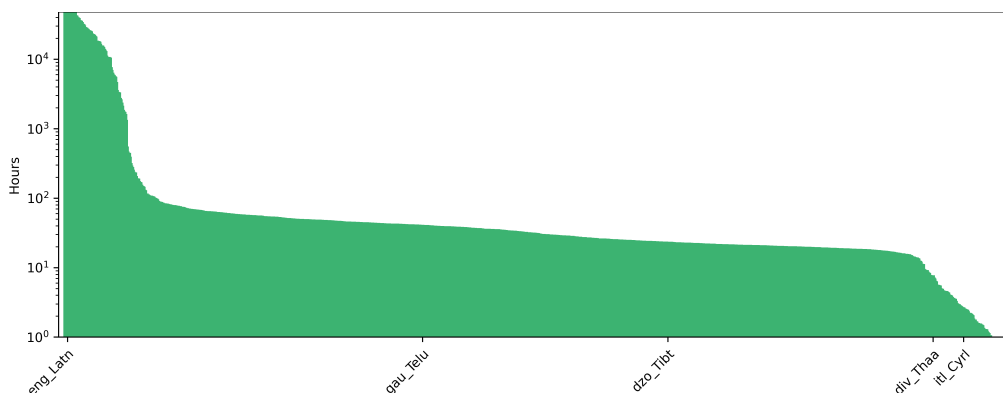


Figure 4 Statistics of the unlabeled data (hours of speech recordings) used to fine-tune Omnilingual ASR for the ASR task.

Due to the heterogeneous nature of the datasets required to represent such a broad spectrum of languages—including variations in recording conditions, speaker demographics, and domain coverage—our development and test data splits are also necessarily heterogeneous. As a result, we caution readers against making direct comparisons between results obtained on different benchmarks. For example, error rates reported on MMS-lab, which features only a handful of speakers per language and contains high-quality recordings, are not directly comparable to those from more diverse datasets such as our own Omnilingual ASR Corpus or the latest spontaneous speech data from Common Voice—which encompass a much wider range of speakers and recording conditions. This is further unpacked and demonstrated in [Section 5.7.4](#).

	Number of hours (total)	Number of languages
Open source datasets	15,000	200
LTPP, internal & licensed data	150,100	1,100
African Next Voices	7,200	13
Open Multilingual Speech Fund	1,940	177
Lanfrica/Naijavoices	110	11
Omnilingual ASR Corpus	3,350	348
Total	120,710	1,690

Table 3 Summary statistics of the training split of the combined ALLASR dataset.

4 Omnilingual ASR Models

This section introduces the Omnilingual ASR models. At a high level, all models follow an encoder-decoder architecture. The speech encoder is a large Transformer (Vaswani et al., 2017) network that extracts high-level cross-lingual representations from input utterances, while the text decoder—either a linear layer or a Transformer decoder—maps these representations into character tokens.

We begin in Section 4.1 by describing how the speech encoder is developed to initialize with strong, massively multilingual speech representations. Section 4.2 then details the creation of our ASR systems, covering both a traditional CTC-based approach and a novel LLM-based approach.

Even with the broad coverage of our supervised ASR models, some languages inevitably remain unsupported. To address this, Section 4.3 introduces a zero-shot extension of our LLM-based models. We show that by providing only a few in-context examples at inference time, the models can perform ASR on previously unseen languages. Section 4.4 further investigates strategies for selecting and constructing these in-context examples to maximize zero-shot performance.

Last but not least, we demonstrate the flexibility of our LLM-based ASR models by repurposing them for speech-to-text translation (S2TT). Remarkably, this requires no dedicated S2TT optimization recipe or complex training pipeline, yet achieves strong performance compared to existing state-of-the-art systems. We detail these results in Section 5.6.

4.1 Massively Cross-Lingual Self-Supervised Representations

At the core of Omnilingual ASR is the speech encoder, whose quality directly determines ASR performance. To ensure that the encoder can extract high-level semantic representations across the wide range of languages we aim to cover, we adopted wav2vec2.0 (Baeovski et al., 2020) for self-supervised learning (SSL), leveraging a large-scale corpus of unlabeled speech. We further scaled wav2vec 2.0 to increase model capacity, enabling it to capture massively multilingual speech representations. We then pre-trained a 7B-parameter wav2vec 2.0 model on 4.3M hours of speech, drawn from a combination of public and internal corpora spanning more than 1,600 languages. To our knowledge, this constitutes one of the largest publicly available SSL model to date, both in terms of parameter count and language coverage. The following sections describe in detail how this was achieved.

4.1.1 Self-supervised Pre-training with wav2vec 2.0

Although first proposed in 2020, wav2vec 2.0 (Baeovski et al., 2020) remains one of the most prominent and effective algorithms for self-supervised learning of speech representations. The basic architecture of wav2vec 2.0 consists of a convolutional feature encoder, a Transformer encoder network, and a quantization module. The convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ maps raw audio $\mathcal{X}$ to a latent representation $Z = (z_1, z_2, \dots, z_T)$, where each z_t here corresponds to 25ms of audio strided by 20ms. The Transformer encoder $g : \mathcal{Z} \mapsto \mathcal{C}$ then processes Z into contextualized representations $C = (c_1, c_2, \dots, c_T)$. In parallel, the quantization module $h : \mathcal{Z} \mapsto \mathcal{Q}$ discretizes Z into $Q = (q_1, q_2, \dots, q_T)$, which are used as learning targets in the objective.

Training proceeds via solving a contrastive task over masked feature encoder output Z . More specifically, spans of time steps in Z are randomly masked, and the objective requires identifying the true quantized latent q_t for a masked time step z_t within a set of distractors sampled from other masked time steps of the same utterance, denoted as $\tilde{q} \in Q$. The loss to minimize is defined as:

$$-\log \frac{\exp(\text{sim}(c_t, q_t))}{\sum_{\tilde{q} \in Q} \exp(\text{sim}(c_t, \tilde{q}))}, \quad (1)$$

where sim stands for cosine similarity, and Q includes 100 distractors and the ground truth q_t itself. Once trained, the quantization module can be discarded, and only the convolutional feature encoder and the Transformer encoder network are required for downstream usage.

4.1.2 Scaling Speech SSL Beyond 2B

Beyond designing effective SSL objectives, model capacity is equally—if not more—crucial to improving representation quality. Since the release of the original 300M-parameter wav2vec2.0 model (Baevski et al., 2020), which at the time was considered large and demonstrated unprecedented success in speech SSL, researchers have pursued two parallel directions: refining SSL algorithms (Hsu et al., 2021; Chen et al., 2022a; Chung et al., 2021; Chiu et al., 2022) and scaling up model size to exploit the potential of ever-larger unlabeled corpora. To date, the largest publicly reported speech SSL models are Google’s Universal Speech Model (USM) (Zhang et al., 2023) and Meta’s XLS-R (Babu et al., 2021), both reaching approximately 2B parameters.

Yet it remains an open question whether 2B parameters marks the effective limit of scaling, either because additional capacity yields diminishing returns, or because 2B parameters are already sufficient for solving most speech tasks. In this work, we revisit the scaling laws of speech SSL by extending wav2vec2.0 from 300M to 1B, 3B, and ultimately 7B parameters. All models are trained on a collection of 4.3M hours of public and internal speech corpora covering more than 1,600 languages (see Section 3.5).

Pre-training Setup

Model	# of layers	model dim	ffn dim	# of attn heads	# params
OmniASR-W2V-0.3B	24	1024	4096	16	317M
OmniASR-W2V-1B	48	1280	5120	16	965M
OmniASR-W2V-3B	60	2048	8192	16	3046M
OmniASR-W2V-7B	128	2048	8192	16	6488M

Table 4 Omnilingual ASR cross-lingual pre-trained wav2vec 2.0 models.

The configurations of our wav2vec2.0 models—including the 300M, 1B, 3B, and 7B variants—are summarized in Table 4. We trained all models using the fairseq2 framework (Balioglu et al., 2023). Because our pre-training data spans many languages and multiple sources, balancing across domains and languages was essential. To this end, we employed a two-step sampling procedure. First, for each data source, we sample the data for the L different languages from a distribution

$$p_l \sim \left(\frac{n_l}{N}\right)^{\beta_L}, \quad (2)$$

where $l = 1, \dots, L$, n_l is the amount of unlabeled audio for each language in the current data source, N is the total amount of unlabeled audio in the current data source, and β_L is the upsampling factor which controls the trade-off between high- and low-resource languages during pre-training. Second, we balanced the different data sources by treating each source as a language and applying the same sampling scheme with a sampling parameter β_D . In practice, we set both β_L and β_D to 0.5.

All our pre-trained models were optimized with Adam (Kingma and Ba, 2014) with a learning rate of $1e-4$, which was warmed up for the first 32K steps followed by polynomial decay to zero for the remainder of training for a total of one million updates. Training batch sizes (measured in hours of audio per batch) were 6, 5.7, 8.5, and 17.6 for the 300M, 1B, 3B, and 7B models, respectively.

4.2 Automatic Speech Recognition

We built on top of the wav2vec2.0 speech encoders described in [Section 4.1](#) to construct two variants of ASR models. The first variant is a connectionist temporal classification (CTC) ([Graves et al., 2006](#)) model, a framework designed to handle input and output sequences of varying lengths without requiring explicit alignments. CTC has become a foundational method in speech recognition and other temporal sequence tasks. By enabling models to learn alignments implicitly, CTC effectively captures temporal dependencies and has driven state-of-the-art performance in multiple applications. Our CTC models comprise of a single linear layer on top of a speech encoder. During training, the speech encoder was seeded from pre-trained wav2vec 2.0, and the entire model was optimized simultaneously using a CTC loss.

Transformer decoders have achieved state-of-the-art performance in natural language processing tasks by effectively modeling complex sequential dependencies. In ASR, stacking a Transformer decoder on top of a speech encoder enables the system to leverage rich acoustic representations while capturing long-range context. This hybrid architecture combines the strengths of speech-specific encoders with the powerful contextual modeling capabilities of Transformers ([Baevski et al., 2021](#); [Radford et al., 2023](#)). As a result, it improves transcription accuracy and robustness in diverse speech recognition scenarios. In the rest of the paper, we refer to this architecture as LLM-ASR, since it uses the same Transformer decoder module commonly found in LLMs. Our LLM-ASR model consists of a speech encoder initialized from a pre-trained wav2vec 2.0 encoder and a Transformer decoder on top of it. The LLM-ASR architecture is depicted in [Figure 5](#).

Formally, both ASR models process a speech segment x through a waveform audio encoder g_s . We denote y as the text transcription sequence corresponding to the speech segment. Our LLM-ASR model additionally holds a text embedding matrix g_t , which maps text tokens and special tokens to vector representations in the Transformer model dimension. The base version of our LLM-ASR model operates on sequences of the form

$$g_s(x) g_t(\langle \text{BOS} \rangle) g_t(y) g_t(\langle \text{EOS} \rangle).$$

where $\langle \text{BOS} \rangle$ and $\langle \text{EOS} \rangle$ denote beginning- and end-of-sequence tokens. This model was then trained using a standard next-token prediction criterion (cross-entropy) to generate the transcription y followed by an end-of-sequence token.

4.3 Zero-Shot Speech Recognition for Unseen Languages

Our supervised ASR models described above support over 1,600 languages using labeled data. However, there remain languages for which no labeled data are available and which therefore cannot be supported by this purely supervised approach. To address this gap, we extend our LLM-ASR model with a zero-shot capability that allows it to perform ASR in any language or domain—including those unseen during training.

The key idea is to shift from single-sample supervision to context-based training. At training time, instead of providing the model with only one speech-text pair, we present $N + 1$ pairs from the same language. The first N pairs serve as context examples and are prepended to the Transformer decoder prompt. The final pair is the target sample, whose transcription the model is trained to predict in the standard next-token prediction framework. This design teaches the model to condition on a few examples of speech-text pairs from a language before producing a transcription for a new utterance in the same language. Because our training corpus covers a large number of languages, we hypothesize that this behavior generalizes to languages absent from training data. As a result, the model acquires a zero-shot ASR capability, effectively enabling communities to extend recognition to their own languages with only a handful of paired examples. The overall architecture of the zero-shot model is illustrated in [Figure 6](#).

In technical terms, we denote the additional N context speech-text pairs as (x_i^c, y_i^c) , where $i \in \{1, \dots, N\}$. Each pair is then embedded with the appropriate modality encoder for the speech and text parts: $g_s(x_i^c)$, $g_t(y_i^c)$. The Transformer decoder then operates on the following sequence syntax:

$$\langle c \rangle \{ \langle cs \rangle g_s(x_i^c) \langle cs \text{ BOS} \rangle g_t(y_i^c) \langle cs \text{ EOS} \rangle \langle /cs \rangle \} \times N \langle /c \rangle g_s(x) \langle \text{BOS} \rangle g_t(y) \langle \text{EOS} \rangle,$$

where $\langle c \rangle$, $\langle /c \rangle$, $\langle cs \rangle$, $\langle /cs \rangle$, $\langle cs \text{ BOS} \rangle$ and $\langle cs \text{ EOS} \rangle$ are special tokens denoting the beginning and end of the context, each context example, and the text part within a context example. Each special token is

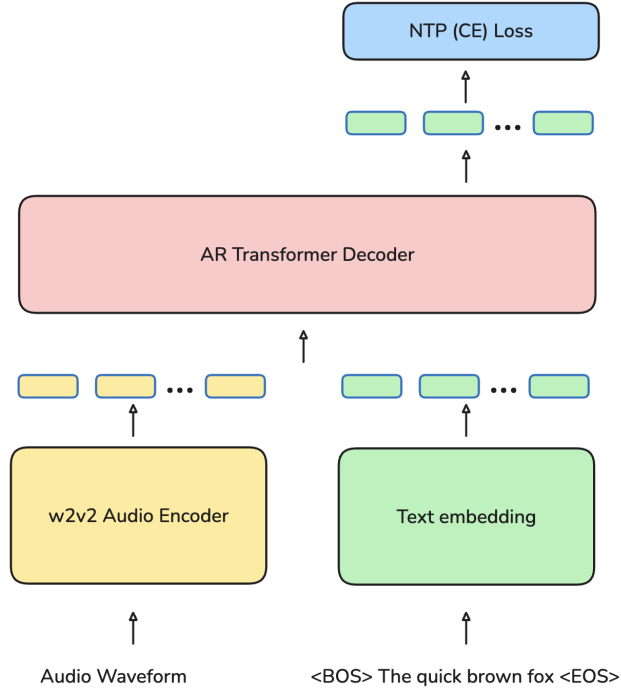


Figure 5 The LLM-ASR model architecture. A wav2vec 2.0 speech encoder and a text embedding matrix embed the speech and text modalities. An autoregressive Transformer decoder emits text tokens, and the system is trained with a next-token prediction objective.

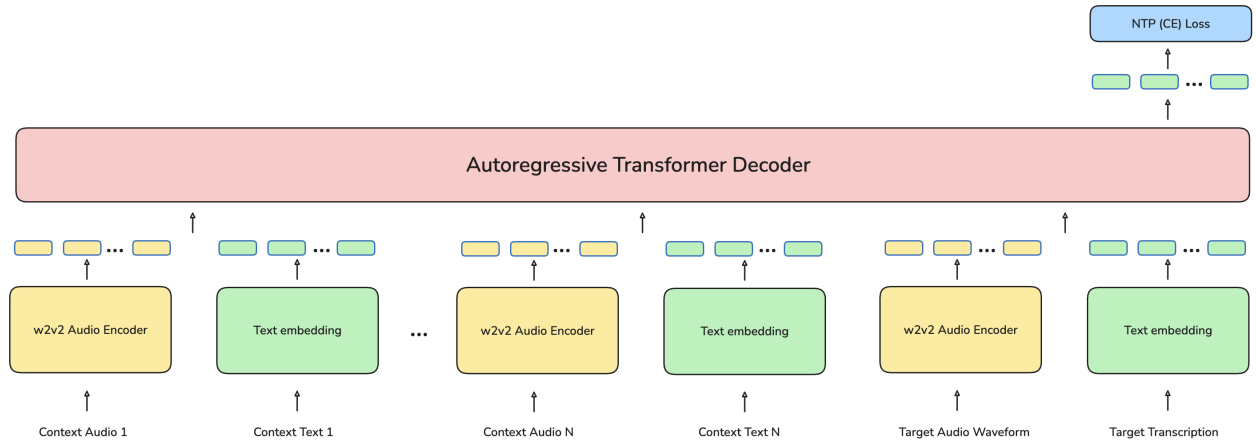


Figure 6 The LLM-ASR model architecture with context examples. Special tokens are omitted for simplicity.

embedded as a text token using g_t , which is omitted above for simplicity of notation. The model was then trained to predict $g_t(y)$ and the final $\langle \text{EOS} \rangle$ using the standard next-token prediction objective. The above sequence syntax, except the last $g_t(y)$ and $\langle \text{EOS} \rangle$, is referred to as the model prompt. At inference time, this prompt is provided and the model generates a candidate transcription $\hat{y}$ and $\langle \text{EOS} \rangle$.

4.4 Selection of Context Examples for Zero-Shot ASR

In [Section 4.3](#), we showed that zero-shot ASR can be performed by providing a few context examples from the target language. At inference time, we have the flexibility to choose which examples to provide, and different construction strategies can significantly impact model performance. Formally, given a target utterance (the query) and a set of candidate speech–transcription pairs (the retrieval base), the task is to select the context examples that maximize transcription accuracy.

As a baseline, examples can be chosen at random within the target language. To improve over this, one natural strategy is to retrieve context examples that are acoustically or semantically similar to the target utterance. A straightforward approach is to embed the target audio into a fixed-length vector and perform nearest-neighbor search within the retrieval base. Prior work on Whisper has shown that kNN-based example selection can improve in-context ASR performance ([Wang et al., 2024a](#)).

For our work, we leverage the SONAR encoder ([Duquenne et al., 2023](#)) as the embedding model to retrieve context examples. SONAR is a multilingual and multimodal system capable of transforming audio or text into a fixed-sized sentence embedding with rich semantic information. In practice, we embedded the target audio sample and used it as the query, while the retrieval base was represented by embeddings of both speech and text. Context examples could then be selected based on nearest-neighbor similarity between the query embedding and the embeddings of the retrieval candidates.

4.5 Conditioning on Language Codes

Multilingual ASR models generally demonstrate the ability to detect the spoken language implicitly and transcribe it correctly ([Pratap et al., 2024](#); [Radford et al., 2023](#)). However, our initial experiments revealed some limitations to this ability. For example, certain languages such as Urdu can be written in multiple scripts, which creates ambiguity for the model. In other cases, closely related languages in the training set may confuse the model about which language to use for transcription. Moreover, in many real-world applications, the user already knows the spoken language in advance and would benefit from being able to provide this information explicitly.

To address these issues, we introduce a mechanism for supplying the model with an additional optional input: a language code together with the desired script. This information is encoded using a dedicated embedding matrix. Specifically, we assign each observed combination of language and script in the training corpus a unique ID, reserving ID 0 to denote an unknown language. During training, this ID—denoted l —is embedded through a matrix g_l . The input sequence to the model becomes

$$g_s(x) \ g_t(\langle \text{language} \rangle) \ g_l(l) \ g_t(\langle \text{BOS} \rangle) \ g_t(y) \ g_t(\langle \text{EOS} \rangle),$$

where $\langle \text{language} \rangle$ is a newly introduced special token. To ensure the model can function both with and without explicit language information, we randomly drop the language input during training with probability p . This enables flexible inference modes: either conditioned on a known language and script or left unconstrained when no prior information is available.

5 Model Training and Evaluation

In this section, we present the training details of Omnilingual ASR models and outline the extensive experiments to validate their capabilities. We begin with the traditional supervised setting in [Section 5.2](#) and [5.3](#). First, we compare Omnilingual ASR with existing large-scale multilingual ASR systems, including WhisperV3 from OpenAI ([Radford et al., 2023](#)), the Universal Speech Model (USM) from Google ([Zhang et al., 2023](#)), and Massively Multilingual Speech (MMS) from Meta ([Pratap et al., 2024](#)), and demonstrate our state-of-the-art

performance on languages overlapping with these existing multilingual systems. We then analyze performance across the full set of 1,600+ supported languages, including more than 500 never before covered by any ASR system.

To extend Omnilingual ASR’s capabilities to support virtually any spoken language, we previously introduced our zero-shot model in [Section 4.3](#). In [Section 5.4](#) and [5.5](#), we show that this model successfully transcribes utterances from languages entirely unseen during training. In [Section 5.6](#), we further adapt the LLM-ASR variant to perform speech-to-text translation with minimal modification, requiring only the insertion of source and target language identifier (LID) tokens into the input sequence. Finally, we present an ablation study on fine-tuning data-mixing ([Section 5.7](#)) and an analysis of the impact of conditioning on language codes ([Section 5.8](#)).

5.1 ASR Training Setup

We trained multilingual ASR models by fine-tuning the pre-trained SSL speech encoders introduced in [Section 4.1](#) using the labeled data described in [Section 3.5](#). For both CTC and LLM-ASR models, we consider four encoder sizes: 300M, 1B, 3B, and 7B parameters. All LLM-ASR variants use the same decoder configuration: a 12-layer Transformer with model dim 4096 and eight attention heads, totaling 1.2B parameters. Throughout, we refer to the LLM-ASR variants by their encoder size.

CTC optimization details. To emit transcriptions, we added a linear layer on top of the pre-trained SSL models, which maps their output to a vocabulary consisting of the set of characters appearing in our labeled training corpus for all languages. We then fine-tuned the entire network with the connectionist temporal classification (CTC) criterion ([Graves et al., 2006](#)). We used Adam ([Kingma and Ba, 2014](#)) with exponential decay rates $\beta_1 = 0.9$, $\beta_2 = 0.98$ to optimize model parameters using a tri-stage schedule: warm-up over the first 10% of updates, hold constant for the next 40%, and exponential decay for the final 50%. All CTC models were trained with a learning rate of 10^{-5} , an effective batch size of 4.2 hours, and for 200k steps.

LLM-ASR optimization details. The LLM-ASR models introduced in [Section 4.2](#) were trained with the same character set described above under a next-token prediction (cross-entropy) objective. Adam was used for those models as well, with a learning rate of 5×10^{-5} and the same β values and learning rate schedule as above. The effective batch size of those models was set to 2.1 hours and the model was trained for 150k steps. At inference time, our LLM-ASR models use beam search decoding with a beam size of five hypotheses.

5.2 Comparison to Other Work

Below, we compare Omnilingual ASR to some of the most prominent existing multilingual ASR work, including Whisper ([Radford et al., 2023](#)), Universal Speech Model (USM) ([Zhang et al., 2023](#)), and Massively Multilingual Speech (MMS) ([Pratap et al., 2024](#)).

5.2.1 Omnilingual ASR vs. Whisper

Whisper is a multilingual speech model trained on approximately 5M hours of weakly labeled web audio and supports a range of speech-processing tasks, including ASR in 99 languages. Its architecture is a Transformer-based sequence-to-sequence model ([Sutskever et al., 2014](#)), consisting of an encoder and a decoder, with the decoder functioning in part like a language model. Thanks to its strong performance and easily accessible API, Whisper has become one of the most widely adopted speech models in the research and developer communities.

In [Table 5](#), we compare Omnilingual ASR models against Whisper’s latest large-v3 release, as well as its smaller variants, using the MMS-Lab ([Pratap et al., 2024](#)), FLEURS ([Conneau et al., 2023](#)), MLS ([Pratap et al., 2020](#)), and Common Voice 22 (CV22) ([Ardila et al., 2020](#)) evaluation sets. We report character error rate (CER) averaged across languages. In this comparison, we only considered languages that Whisper covers in each benchmark; the number following each dataset name indicates the corresponding number of languages evaluated. To further strengthen the comparison, we also trained n-gram language models for FLEURS and MLS languages using their training transcripts, and considered LM fusion with those models for our

Model	MMS-Lab-66		FLEURS-81		MLS-8		CV22-76		Win Rate	
	dev	test	dev	test	dev	test	dev	test	$n = 81$	$n = 34$ (top 50)
<i>Prior Work</i>										
Whisper small	66.8	64.3	51.5	50.8	6.2	4.9	103.6	111.7	-	-
Whisper medium	55.5	54.5	48.0	47.8	6.8	4.6	79.8	87.9	-	-
Whisper large-v3	32.0	30.9	22.0	22.6	2.3	2.0	27.3	55.6	-	-
<i>This Work</i>										
300M CTC	4.9	4.7	11.7	11.8	4.6	4.1	16.7	17.6	37	-
1B CTC	3.0	2.8	8.5	8.6	3.3	3.1	13.5	14.8	48	-
3B CTC	2.2	2.0	7.7	7.8	3.1	2.7	12.3	13.7	54	-
7B CTC	1.9	1.7	7.2	7.3	2.8	2.5	11.6	13.8	61	-
300M LLM-ASR	1.7	1.9	8.0	7.8	3.6	3.2	6.5	7.1	46	-
1B LLM-ASR	1.4	1.2	6.7	6.6	2.9	2.7	5.9	6.5	55	-
3B LLM-ASR	1.3	1.1	6.3	6.2	2.8	2.6	6.3	6.6	57	-
7B LLM-ASR	1.1	1.0	5.9	5.6	2.5	2.4	5.5	6.4	65	24
7B LLM-ASR + LM	-	-	5.7	5.5	2.5	2.4	-	-	65	-

Table 5 Comparison against Whisper v3, including its large (1.5B), medium (769M), and small (244M) variants. For each benchmark, we report average CER across languages on both dev and test splits. The comparison only considers languages that Whisper covers in each benchmark, and the number that follows the dataset name indicates the number of languages considered. The two rightmost columns show the win rate of our model against Whisper large v3 on the FLEURS test set: $n = 81$ considers the entire FLEURS-81 languages, while $n = 34$ only considers the top 50 most spoken languages in the world that are covered by FLEURS (34 of them).

largest variant using hyperparameters optimized on the dev set. The main results from this comparison are summarized in Table 5.

More specifically, we find that even our smallest model outperforms Whisper large-v3 on most evaluation sets, as measured by average CER across languages. Our 300M-CTC variant surpasses Whisper-large on MMS-Lab-63, FLEURS-82, and CV22-76, and falls behind only on MLS-8. As we scale encoder size, the gap with Whisper on the former three benchmarks continues to widen. Against Whisper small and medium, the 300M-CTC outperforms them on all four benchmarks.

Moreover, Omnilingual ASR performs strongly on the world’s most spoken languages while supporting long-tail ones. Whisper shows strength on some of the highest-resource languages, as reflected in its MLS-8 results, likely due to the large amount of labeled training data in those languages. However, its accuracy drops sharply on long-tail languages included in other benchmarks. Our models, on the other hand, while remaining strong on high-resource languages, outperform significantly on long-tail languages. In general, we find that the Whisper models’ average CER across languages is disproportionately affected by a long set of poorly supported languages. To provide additional insights to the comparisons, Table 5 reports the number of languages on which our models outperform Whisper large-v3 on FLEURS-81, including a breakdown for the 34 of the world’s 50 most spoken languages⁷ that are covered in FLEURS-81. Comparing our 7B-LLM against Whisper large-v3, we achieve an 80% win rate (65 out of 81) across all languages in FLEURS-81, and 71% (24 out of 34) on the most spoken languages.

Finally, comparing our own variants, the LLM models consistently outperform their CTC counterparts by a wide margin. Error analysis shows that CTC models often fail due to script misprediction: when the wrong script is chosen for an input utterance, the decoded characters belong to another language altogether. This issue is particularly common in low-resource settings as models are less familiar with their scripts. By contrast, our LLM-ASR models benefit from the ability to condition on language codes at inference time (while still working without them), which largely resolves the wrong-script problem. The LLM results in Table 5 are reported with language conditioning. Ablations on language conditioning are presented in Section 5.8.

⁷<https://www.ethnologue.com/insights/ethnologue200>

5.2.2 Omnilingual ASR vs. USM

USM and Omnilingual ASR follow a broadly similar development recipe: both begin with large-scale self-supervised pre-training of a Transformer encoder, followed by appending a decoder on top and fine-tuning the entire model with labeled data. In USM’s case, the encoder adopts a Conformer architecture (Gulati et al., 2020), a convolution-augmented Transformer variant. Pre-training is performed with the BEST-RQ algorithm (Chiu et al., 2022) on roughly 12M hours of proprietary YouTube audio spanning 300 languages, and fine-tuning for ASR is carried out on 90K hours of labeled data across 100 languages. The Conformer encoder itself has 2B parameters, and the decoder is an RNN-Transducer that has a built-in neural language model. Additional USM variants (e.g., USM-M and USM-M-adapter) extend this setup with multi-stage pre-training pipelines that include text pre-training and labeled audio, totaling about 20K hours. In contrast, Omnilingual ASR encoders are pre-trained solely on unlabeled speech data.

Model	FLEURS-102	
	dev	test
<i>Prior Work</i>		
Maestro-U (Chen et al., 2022b)	-	8.7
USM	-	6.9
USM-M	-	6.5
USM-M-adapter	-	6.7
<i>This Work</i>		
7B CTC	7.4	7.5
1B LLM-ASR	7.3	7.2
3B LLM-ASR	6.8	6.7
7B LLM-ASR	6.4	6.2
7B LLM-ASR + LM	6.2	6.1

Table 6 Comparison against USM and its variants on FLEURS-102. We report average CER across languages. For USM and its variants, only test set results are available; we report our results on both dev and test splits.

Since USM and its variants are not publicly accessible, we rely on their reported results on FLEURS-102, presented in Table 6. We see that when considering the full FLEURS-102 benchmark (as opposed to FLEURS-81 in Table 5), our 7B-LLM model still outperforms 7B-CTC. Compared to the best USM variant (USM-M), which achieves a CER of 6.5%, our 7B-LLM achieves 6.2%, and when we incorporate LM fusion at inference, the CER is further reduced to 6.1%. Despite the fact that our models are pre-trained on more than 50% less unlabeled speech data than USM (4.3M vs. 12M hours) and do not adopt a sophisticated pre-training pipeline involving multiple stages (as USM does), our models still outperform the USM models. We largely attribute this to the impact of encoder size scaling.

5.2.3 Omnilingual ASR vs. MMS

Similar to USM and Omnilingual ASR, MMS (Pratap et al., 2024) takes advantage of SSL to leverage large quantities of unlabeled speech data to pre-train a Transformer encoder so as to initialize it with rich cross-lingual speech representations, before appending a decoder and fine-tuning the entire model with labeled data. Specifically, MMS uses wav2vec 2.0 (Baeovski et al., 2020) to train a 1B Transformer encoder network, leveraging around 500k hours of unlabeled speech data and covering approximately 1400 languages. After appending a linear layer as a decoder to the pre-trained encoder, the entire model is fine-tuned with around 45k hours of labeled data to cover ASR for approximately 1100 languages using CTC.

For FLEURS-102, MMS incorporates a sophisticated fine-tuning pipeline to optimize its ASR performance—the Transformer encoder is modified with adapter modules (Houlsby et al., 2019), where a different set of adapter weights is used for each language. Specifically, MMS has an adapter module augmented to every layer of its Transformer encoder, where the adapter is added after the last feed-forward block. Each adapter module consists of a LayerNorm layer, a downward linear projection, followed by a ReLU activation, and an upward linear projection. After an initial fine-tuning stage across all languages, MMS performs a second stage of

language-specific fine-tuning. In this step, the model introduces a randomly initialized linear layer that maps to the output vocabulary of a language, alongside a dedicated language-specific adapter. These additional parameters are then fine-tuned on the labeled data available for that language.

Model	MMS-Lab-1143	FLEURS-102	MLS-8
<i>Prior Work</i>			
MMS - single-domain training + LM	-	6.4	8.7
MMS - multi-domain training + LM	2.1	6.3	9.0
<i>This Work</i>			
7B LLM-ASR	1.9	6.2	8.0
7B LLM-ASR + LM	-	6.1	8.0

Table 7 Comparison against MMS on the test sets of MMS-Lab-1143, FLEURS-102, and MLS-8. We report average CER across languages except for MLS-8, where we report WER. “MMS - single-domain training” means the MMS model is fine-tuned on just that particular dataset, and “MMS - multi-domain training” means the model is fine-tuned on the full 45k hours of MMS labeled data. Both reported MMS results are with n-grams LM decoding.

We compare MMS with Omnilingual ASR in Table 7, reporting CER on MMS-Lab-1143 and FLEURS-102, and WER on MLS-8. The results are averaged across all the languages in the corresponding datasets. “MMS - single-domain training” means that the MMS model is fine-tuned on just that particular dataset, while “MMS - multi-domain training” means the MMS model is trained on their entire 45k hours of labeled data. After training, during inference time, MMS uses an n-gram model trained on Common Crawl for better decoding results. From the table, we see that our 7B-LLM outperforms MMS on all evaluation sets, regardless of the setting for which MMS models are optimized.

5.3 Evaluation on 1600+ languages

In the previous section, we compared Omnilingual ASR with Whisper, USM, and MMS, showing that our models set or match state-of-the-art performance across existing multilingual benchmarks. We now turn to a broader analysis of Omnilingual ASR’s performance on the full set of 1,600+ languages it supports—including more than 500 languages that have never before been covered by any ASR system.

Evaluating models at this scale requires a structured approaches. As such, we adopted two complementary protocols: (i) dividing languages into high-, mid-, and low-resource categories based on the amount of labeled training data available, and (ii) sorting languages into 14 major groupings following the principles outlined below. For simplicity, all test splits are aggregated by averaging results across languages within each category of the respective evaluation protocol.

5.3.1 Evaluation based on Resource Buckets

	High	Mid	Low
# of lang in this bucket	249	881	546
7B-CTC	3.7 ± 0.7	4.4 ± 0.6	18.6 ± 1.2
7B-LLM	3.13 ± 0.7	3.0 ± 0.3	18.0 ± 1.2

Table 8 Mean CER for each language-resource bucket with 95% Confidence Intervals. High-resource languages have >50 hours training data, mid-resource have between 10-50h, and low- have <10h. Both models do not employ LM fusion.

We group languages into resource buckets according to the amount of labeled training data available in ALLASR. High-resource languages are those with more than 50 hours of training data, mid-resource languages fall between 10–50 hours, and low-resource languages have fewer than 10 hours. This results in 249, 881, and 549 languages in the high-, mid-, and low-resource buckets, respectively. To ensure a sufficient validation signal, we exclude languages with less than 30 minutes of data in their validation splits.

	High	Mid	Low
# of lang in this bucket	249	881	546
7B-CTC	231	823	184
7B-LLM	236	841	195

Table 9 Number of languages within each resource-bucket where our models obtain CERs below 10.

Table 8 reports the mean CER across languages in each bucket, while Table 9 shows the number of languages achieving $\text{CER} < 10$ within each bucket. Both of our models can achieve low CERs (under 5) in the high- and mid-resource categories, with 90% of languages in these buckets meeting this threshold. On the low-resource bucket, where we have less than 10 hours of training data per language, the percentages of languages that meet the CER quality threshold fall to 34% and 36%, with an average CER of 18.6 and 18.0 for 7B-CTC and 7B-LLM, respectively. In Section 5.7.5, we examine the performance of long-tailed languages and provide a recipe for further fine-tuning our models on specific languages to achieve optimal performance.

5.3.2 Evaluation based on Language Groupings

Grouping	# of lang	Avg CER	$\text{CER} \leq 10$	%
Afroasia	92	11.8	61	66%
Amazbasi	83	2.0	82	99%
Amerande	67	2.0	66	99%
Atlacong	389	9.3	280	72%
Austasia	35	5.4	31	89%
Austrone	239	5.1	193	81%
Caucasus	35	3.9	35	100%
Dravidia	22	7.3	18	82%
Indoeuro	209	9.1	154	74%
Mesoamer	159	7.8	115	72%
Newguine	77	5.5	63	82%
Nilosaha	56	4.4	50	89%
Norameri	42	4.8	37	88%
Sinotibe	65	8.2	52	80%
Total	1570	7.1	1237	78%

Table 10 Average CER across languages under 14 language groupings using our 7B-LLM model without LM fusion. We only considered languages that can be classified into one of the 14 groupings and dropped the rest of the languages our models support. # of lang denotes the number of languages belonging to that particular grouping covered in our evaluation sets. $\text{CER} \leq 10$ indicates the number of languages belonging to that grouping that achieves a CER no greater than 10, and % shows the percentage of that.

The main principles used for grouping are as follows. Languages are first grouped according to their respective families; the definition of the term *family* follows the linguistic genealogy research in Hammarström et al. (2024). In cases where family-based grouping does not yield a large enough number of group members (i.e., for either small families or families with a small number of members being represented in our datasets, as well as for language isolates), languages are additionally grouped by linguistic proximity. Although the eight-letter labels used for those groups (e.g., Caucasus, Norameri, Amerande) may sound geographical, linguistic proximity is not to be understood solely as geographical proximity but also as typological proximity (i.e., following aspects of linguistic typology). The grouping resulted in 14 groups of different sizes, ranging from 389 members for the largest group to 22 members for the smallest one.

In Table 10, we present the results of our 7B-LLM model across the 14 language groupings. We omit languages our models support but cannot be classified into one of the 14 groupings in this analysis. # of lang’ denotes the number of languages under that particular grouping that are covered in our evaluation sets, and Avg CER

shows the average CER across languages under that grouping. Additionally, in order to get a broader sense of quality, we measure the number of languages for which $\text{CER} \leq 10$. This indicates how many languages the model produces, on average, no more than one error in ten characters. While this measure is very coarse, it enables us to get a sense of quality across such a large number of languages. From the table, we see that overall our model meets the CER quality threshold for 78% of the 1570 languages we evaluate on, and is able to reach a CER below 10 for all groupings except for Afroasia, for which we get 11.8.

By measuring our model’s performance through the lens of resource buckets and language groupings, our analysis in [Section 5.3](#) demonstrates our models’ ability to transcribe a massive variety of languages while maintaining reasonable to high quality.

5.4 Accuracy of Zero-Shot Models on Unseen Languages

We conducted experiments to evaluate the generalization of our zero-shot ASR model described in [Section 4.3](#) to unseen languages. To that end, we excluded a set of 32 languages from our training set, which will be used for evaluation. The set of evaluation languages was chosen at random but in a manner that asserts that half of the languages are high-resource languages that are represented in more than one evaluation set, and the other half are low-resource languages that may only appear in a single evaluation set. Since some evaluation sets contain only a small number of the evaluation languages, it does not make sense to report accuracy by evaluation set in this setting. Instead, for each evaluation language, we compute its overall CER across all evaluation sets, and average this number across languages. The context examples were chosen randomly for each utterance from the same dataset and in a consistent manner across models.

The zero-shot models are compared to a CTC and LLM-ASR baselines, both trained excluding the same set of languages, which are then used for evaluation. To find an optimal setting for generalizing to unseen languages, we experimented with a number of variants of the zero-shot model. The candidates vary by the number of context examples used, the seed used to initialize the speech encoder, and whether the speech encoder was frozen during that training or not. Results appear in [Table 11](#). From the table, we see that among baselines, the CTC model generalizes better to unseen languages than the LLM-ASR variant. However, when augmented with conditioning on context examples, the LLM-ASR model outperforms the CTC model and reduces the overall CER on unseen languages from 26.33% to 14.4% using a context size of 10, the largest context size we experimented with. Among zero-shot models, we found that seeding from CTC reduces the generalization ability to unseen languages. We also observed that tuning the speech encoder was crucial for demonstrating the zero-shot ability in a manner superior to baseline models.

An additional observation is that zero-shot models somewhat degrade accuracy on some datasets of seen languages compared to their non zero-shot counterparts. However, we release separate models for stronger support in the languages appearing in our training set, making this metric less important for zero-shot models. Two exceptions are the FLEURS-102 and CV22 datasets, in which zero-shot models outperform the baseline models. The reason for this is a relatively high number of utterances in those datasets where the script is being misrecognized by non zero-shot models, thus vastly increasing the CER. As zero-shot models are provided with a number of context speech and transcription pairs from the language, they significantly reduce script and language confusion errors.

Reference text:	was kommt als nächstes
CTC:	vas comt als nekstes
LLM-ASR:	vas komt als nekstes
Few-Shot LLM-ASR:	was komt als nächstes

Figure 7 A German example of the zero-shot model (German was excluded from training of this model). While baseline models struggle with the correct spelling, the zero-shot ASR model produces a more accurate hypothesis.

One example of the superiority of zero-shot models on unseen languages can be seen in [Figure 7](#). This illustrates an example in German, which was excluded from training in all models in this subsection. While non zero-shot models make considerable spelling errors, zero-shot models do visibly better.

Model	Context	Unseen	MMS-Lab	Omnilingual ASR Corpus	FLEURS -102	MLS	CV22
CTC	0	26.3	4.2	23.1	8.5	2.7	15.4
LLM-ASR	0	31.0	2.9	20.3	7.6	2.7	15.5
ZS LLM-ASR, CTC seed	5	19.3	3.4	21.2	6.8	2.9	9.2
ZS LLM-ASR, CTC seed, Fr.	5	26.5	4.0	23.2	8.0	2.7	11.8
ZS LLM-ASR, w2v2 seed	5	17.6	3.7	21.9	7.1	2.9	8.7
ZS LLM-ASR, w2v2 seed	10	14.4	4.3	23.2	8.3	3.1	10.3

Table 11 Generalization to unseen languages of the zero-shot models. Unseen refers to the language average CER across all evaluation sets for unseen languages. The rest of the evaluation sets specified refer to the portion of those sets with languages seen during training.

5.5 Constructing Context Examples for Zero-Shot ASR

In this section, we present a series of selection approaches for studying how the model uses context in the zero-shot ASR setting. Limited by the language coverage of the SONAR speech encoder, we trained another LLM-ASR with five context examples but with a different set of 32 holdout languages (supported by SONAR). We did not condition on language codes for this setting. The holdout languages remain diverse, encompassing languages with distinct scripts and belonging to various language groupings. Our set of holdout languages includes some very high resource languages, such as English and Spanish; most of the languages are mid-resource, ranging from 100-300 hours in the entire training corpora, and also a few lower resource languages below 100 hours, such as Welsh and Marathi. The model architecture and training basically follow [Section 5.4](#). We initialized the speech encoder with the 7B wav2vec 2.0 encoder, and the speech encoder was updated during ASR training. After training, we evaluated zero-shot ASR performance on the holdout languages. For each evaluation set, we selected context examples from the corresponding training set for all selection approaches.

Intuitively, one strategy is to provide context examples that share similarities with the target; another is to sample a diverse set of context examples, where we try to cover as much variety of the unseen language as possible. An open question is which features to use when selecting context examples—textual, semantic, or audio similarity. These features are not entirely independent (e.g., higher semantic similarity can also lead to higher text overlap). In this section, the baseline approach would be randomly selecting context examples from the retrieval base without duplicates, and the random baseline, to some extent, would consist of diverse context examples of different aspects.

For selecting context examples that are similar to the target, we focused on these three features: text, semantic, and audio. For semantic-based selection, we used SONAR speech embedding as a query to retrieve examples from the SONAR speech embeddings (sonar_ss), and from the SONAR text embeddings in the retrieval base (sonar_st) using nearest neighbors based on the embedding cosine similarity.

For audio-based similarity, we utilized embeddings derived from SSL representations for selection. We extracted frame-level audio representations using a pre-trained-only wav2vec-2.0 encoder and then mean-pooled the frame-level representations into a single embedding vector for utterance retrieval (w2v2), employing cosine similarity between embeddings. The embeddings obtained from wav2vec 2.0 representations may be more phonetic than semantic ([Choi et al., 2024](#)) compared to SONAR embeddings. For text-based similarity, we performed a similarity search based on bm25 ([Robertson and Zaragoza, 2009](#)) to select context examples, where we used the target transcript as query (text_sim) in this case. Note that the text-based similarity baseline cannot be fairly compared to the random selection baseline, as it involves using the target transcript for searching. For selection methods based on similarity to the target, the selected context examples were placed in the order of increasing similarity.

We now turn to the alternative method for constructing context examples based on text in the retrieval base. In this approach, we selected five examples with the highest unique bigram counts of characters from the retrieval base (bigram), and the same five examples were provided as context examples for all testing audio samples. The bigram selection method maximizes textual diversity within context examples, contrasting with

other selection methods that aim to maximize similarity to the target audio. However, the bigram selection method would be biased towards selecting longer context examples, as we did not impose any constraints on the total context length.

For sanity checks and for understanding the capability of the LLM-ASR model, we provided the model with the “answer,” setting all five context examples to <target audio><target text> (same_ex). In this approach, we expect to see significantly improved accuracy compared to all other baselines.

The results averaged on all holdout languages are shown in Table 12. We consider text_sim and same_ex as oracle approaches, as the target transcript is used. Using SONAR embeddings to select examples (sonar_ss and sonar_st) yields lower UERs compared to the random selection baseline, reducing CER by up to 11.2% relative. Using speech-to-speech or speech-to-text embedding retrieval does not show much difference, allowing flexibility to retrieve from either text or speech embeddings. Using wav2vec 2.0 mean-pooled embeddings for selection does not show obvious improvements over the random baseline.

The bigram selection yields only a slight improvement over the random baseline, suggesting that the model may struggle to effectively learn from context examples that are not directly related to the target. Moving to the oracle results, having context examples with higher text similarity to the target (text_sim) shows further gains compared to the SONAR selection baseline. The stronger oracle approach of providing the model with the target audio and transcript pair as context examples (same_ex) significantly reduces the UER.

From the above results, we can see that even though the model was trained on randomly selected context examples, how we constructed context examples during inference can significantly influence the transcribed text in the zero-shot setting. The oracle results corroborate the fact that the LLM-ASR model can make use of the context examples. From the baseline results, we observe that the model benefits more from examples similar to the target sample over mere textual diversity among context examples.

We present an example of how the transcribed text of the same sample changes with different selection methods in Table 13.

	random	sonar_ss	sonar_st	w2v2	bigram	oracle	
						text_sim	same_ex
MMS-lab	17.9	15.9	16.3	17.4	17.4	15.3	11.6
FLEURS	24.4	23.5	23.6	24.0	24.1	23.1	16.4
CV	18.6	17.5	17.1	18.5	17.9	16.1	9.8

Table 12 Results for the difference methods of context examples selection. The numbers stand for average UER on the holdout languages.

reference text	the school also encourages its students to participate in extracurricular activities via various programmes
random	the school also encuriges it stoedents to partisipate in ekstra curricular activities wia waries programs
sonar_ss	the school also encuriges its students to partisipet in extra curricular activities via veries programs
same_ex	the school also encouriges its students to participate in extracurricular activities via various programmes

Table 13 An example of the transcribed text with different selection methods. English is excluded in the training for this model. Some spelling can be potentially corrected by just changing the context examples provided at inference time.

5.6 Applications to Speech-to-Text Translation

As mentioned at the start of Section 5, we adapted the LLM-ASR variant to perform speech-to-text translation (S2TT) with minimal modification, requiring only the insertion of source and target language identifier (LID) tokens into the input sequence. Despite this simplicity, our experiments show that the model consistently

outperforms Whisper and other baselines. Moreover, its performance is comparable to the state-of-the-art SeamlessM4T (SEAMLESS Communication Team, 2025), which employs a more complex development pipeline specifically designed for speech translation.

5.6.1 S2TT Experimental Setting

We first evaluate translation directions of X to English, denoted as X-Eng. For this setting, we used CoVoST2 (Wang et al., 2020) and FLEURS (Conneau et al., 2023) as benchmarks—CoVoST2 covers 21 source languages, while FLEURS spans 101. Our main comparisons are against Whisper and SeamlessM4T v1.

We reused a large proportion of the X-Eng training data from the SeamlessM4T project. Following the setup in SeamlessM4T, we do not include FLEURS samples in the training data so that they can serve as a reliable measure of out-of-domain performance. We consider OmniASR-W2V- $\{1B, 3B, 7B\}$ as the encoder when constructing our S2TT models. Consistent with our LLM-ASR model in Section 5.1, the decoder is a 1.2B-parameter Transformer in a decoder-only configuration, and we reused the same hyperparameters for training our S2TT models.

5.6.2 S2TT Results and Discussion

Model	Model Size	CoVoST2 21-Eng	FLEURS 81-Eng	FLEURS 101-Eng
<i>Prior Work</i>				
XLSR-2B-S2T (Babu et al., 2021)	2.6B	22.1	-	-
Whisper Large v2	1.5B	29.1	17.9	-
SeamlessM4T v1 Medium	1.2B	29.8	20.9	18.4
SeamlessM4T v1 Large	2.3B	34.1	24.0	21.4
AudioPaLM-2-8B-AST (Rubenstein et al., 2023)	8.0B	37.8	19.7	-
<i>This Work</i>				
OmniASR-LLM-1B	2.2B	34.6	19.1	16.7
OmniASR-LLM-3B	4.3B	36.7	22.1	19.4
OmniASR-LLM-7B	7.7B	37.1	23.5	20.8

Table 14 Omnilingual ASR S2TT results in comparison to state-of-the-art speech translation models. We report average BLEU (higher is better) scores across all X-Eng directions on CoVoST2 and FLEURS test splits. Model size indicates the # of params of that particular model. For Whisper, we started with v3, but its average performance was worse than v2, hence we compared against v2 here.

Results are presented in Table 14, where we also include several baselines in addition to Whisper and SeamlessM4T. For both CoVoST2 and FLEURS, we report the average BLEU scores across all X-Eng directions on their test sets. Since Whisper only covers 81 out of the 101 to English directions in FLEURS, we also evaluated our models only on these 81 languages to produce a fair comparison against Whisper.

We see that our models largely outperform Whisper on both CoVoST2 and FLEURS, regardless of the model size. Considering individual language results, we find that our model beats Whisper on 74 out of 81 X-Eng directions on FLEURS. Compared to SeamlessM4T, our best model outperforms its medium variant across the board, but slightly lags behind its large variant on FLEURS-81 by 0.5 BLEU score point and 0.6 on FLEURS-101. Note that SeamlessM4T initialized its decoder with a pre-trained decoder from NLLB (NLLB Team, 2024), whereas here we trained our decoder from scratch without any pre-training. the decoder is a 1.2B-parameter Transformer in a decoder-only configuration.

5.7 Impact of Datamix

Beyond our primary goal, which is to maximize support for low-resource languages while minimizing regressions in higher-resource ones, we also sought to build robustness against the wide range of noise conditions and speaker variability found in real-world audio. To meet these dual objectives, we designed a series of ablations and

upsampling experiments tailored to the challenges of our ALLASR dataset, which is both highly heterogeneous in audio quality and heavily imbalanced in language coverage.

5.7.1 Upsampling Low-Resource Languages

We upsampled at both the corpus- (datasource) and language-levels according to the following hyperparameters: β_c and β_l . β_c determines the relative weight assigned to a particular corpus, and β_l determines the relative weight for a particular language within a corpus. More precisely, for each corpus, we sampled language L according to $p_l \sim \left(\frac{n_l}{N}\right)^{\beta_l}$, where $l = 1, \dots, L$ is the language, n_l is amount of labeled ASR data for each language within the corpus, and N is total volume of data in the dataset. Sampling across corpora was determined by treating each corpus as a language in the above equation, and using parameter β_c . This approach is consistent with previous work (Pratap et al., 2024). Lower beta values result in higher levels of upsampling of smaller data sources, with 0.0 causing uniform sampling across languages (irrespective of the amount of training data available for each language), and 1.0 representing a baseline where we simply concatenate all data without performing any upsampling.

To determine the optimal upsampling hyperparameters, we performed a sweep across different combinations of β_c and β_l . For hyperparameter selection, we trained a 1B CTC model for 200K steps, and then compared results on all three evaluation protocols described in Section 5: resource-based (Table 17), language-family (Table 16), and corpus-based (Table 15).

Looking at Table 17, we can see that as we increase language-level upsampling (ie, decrease β_l at a given β_c), CERs decrease for low-resource languages. The baseline (1.0, 1.0) setting, which corresponds to no upsampling, performs by far the worst on low-resource languages. According to results on the resource-based protocol, the best setting is (0.0, 0.0), which is maximal (uniform) upsampling at both the corpus- and language-levels. This setting also gives the highest performance according to the language-grouping evaluation protocol, producing the lowest CERs within each grouping (see Table 16).

Table 15 shows results on the corpus evaluation protocol. Here, the (0.5, 0.25) setting achieves best results in the corpus evaluation protocol. We can also see here that the (0.0, 0.0) setting obtained lowest CERs on MMS-lab corpus—which comprises over 1000+ languages. This helps explain why it performed so well on the language-based evaluation protocols: they are largely determined by the broad language coverage of MMS-lab. However, this increased MMS-lab performance came at the expense of other datasets such as Babel and CV22, which are known to contain noisier audio data and more diverse speaking conditions. As described subsequently in Section 5.7.2, over-indexing on the narrow audio domain of MMS-lab can have adverse effects on model robustness. Consequently, we chose the (0.5, 0.25) setting when training our final OmniASR models, as this performs well across all corpora and still achieves good results on the language-based protocols.

Condition	Babel	MMS-lab	CV22	FLEURS_102	MLS	OmniASR	Avg
cbeta_0.0_lbeta_0.0	27.55	4.47	17.73	9.64	3.86	24.08	14.55
cbeta_0.25_lbeta_0.5	25.05	7.07	16.74	9.24	3.25	25.75	14.52
cbeta_0.5_lbeta_0.5	25.71	6.32	17.14	9.63	3.26	26.23	14.71
cbeta_0.75_lbeta_0.5	27.01	5.82	17.10	10.46	3.32	27.54	15.21
cbeta_0.5_lbeta_0.25	25.41	6.05	16.42	9.42	3.32	26.08	14.45
cbeta_0.5_lbeta_0.75	25.85	6.55	17.94	9.61	3.20	26.35	14.92
cbeta_1.0_lbeta_1.0	28.72	6.09	21.30	11.15	3.20	29.33	16.63

Table 15 Performance (CER) across dev splits for each corpus in AllASR dataset for different *beta* values. The rightmost column (avg) is separated for clarity.

5.7.2 Generalizing to Unseen Audio Distributions

In addition to optimizing for low-resource languages, we also wanted to ensure our model was robust to various audio conditions. As such, we ran an ablation where we trained models on the ALLASR dataset, holding out one corpus at a time. Here *AllASR* refers to: MMS-lab, Omnilingual ASR Corpus, OMSF, FLEURS-102,

CERs for (cbeta, lbeta) upsampling							
Language Groupings	(0.0, 0.0)	(0.25, 0.5)	(0.5, 0.25)	(0.5, 0.5)	(0.5, 0.75)	(0.75, 0.5)	(1.0, 1.0)
Afroasia	16.35 $\pm$ 3.11	18.70 $\pm$ 3.20	18.06 $\pm$ 3.18	18.45 $\pm$ 3.40	18.86 $\pm$ 3.94	17.96 $\pm$ 3.20	19.37 $\pm$ 4.12
Amazbasi	3.12 $\pm$ 0.41	5.04 $\pm$ 0.58	4.34 $\pm$ 0.52	4.39 $\pm$ 0.52	4.48 $\pm$ 0.55	4.13 $\pm$ 0.51	4.12 $\pm$ 0.57
Amerande	3.40 $\pm$ 0.72	4.95 $\pm$ 0.85	4.45 $\pm$ 0.85	4.55 $\pm$ 0.87	4.63 $\pm$ 0.88	4.48 $\pm$ 0.96	4.65 $\pm$ 1.13
Atlacong	12.43 $\pm$ 1.19	15.47 $\pm$ 1.15	14.70 $\pm$ 1.19	14.90 $\pm$ 1.19	15.05 $\pm$ 1.19	14.95 $\pm$ 1.25	15.62 $\pm$ 1.34
Austasia	11.80 $\pm$ 6.55	14.41 $\pm$ 6.32	12.88 $\pm$ 7.14	13.52 $\pm$ 7.14	13.98 $\pm$ 7.37	13.43 $\pm$ 7.29	14.19 $\pm$ 6.90
Austrone	6.63 $\pm$ 1.10	8.07 $\pm$ 1.11	7.76 $\pm$ 1.14	7.88 $\pm$ 1.14	7.99 $\pm$ 1.15	7.99 $\pm$ 1.20	8.35 $\pm$ 1.25
Caucasus	11.89 $\pm$ 4.39	11.95 $\pm$ 3.46	13.62 $\pm$ 5.10	12.58 $\pm$ 4.33	13.76 $\pm$ 5.24	13.13 $\pm$ 5.39	14.43 $\pm$ 5.16
Dravidia	13.16 $\pm$ 8.77	14.26 $\pm$ 7.57	13.73 $\pm$ 7.89	14.02 $\pm$ 7.73	14.02 $\pm$ 7.15	13.99 $\pm$ 7.80	14.08 $\pm$ 6.70
Indoeuro	13.15 $\pm$ 1.95	14.47 $\pm$ 1.99	14.35 $\pm$ 2.00	14.74 $\pm$ 2.02	15.15 $\pm$ 2.15	15.15 $\pm$ 2.09	17.25 $\pm$ 2.28
Mesoamer	10.53 $\pm$ 1.93	13.05 $\pm$ 1.88	12.36 $\pm$ 1.93	12.55 $\pm$ 1.92	12.67 $\pm$ 1.91	12.59 $\pm$ 1.99	13.21 $\pm$ 2.09
Newguine	7.27 $\pm$ 2.31	9.24 $\pm$ 2.39	8.68 $\pm$ 2.44	8.83 $\pm$ 2.44	8.97 $\pm$ 2.45	8.76 $\pm$ 2.55	9.07 $\pm$ 2.65
Nilosaha	7.23 $\pm$ 1.81	10.36 $\pm$ 1.76	9.25 $\pm$ 1.85	9.46 $\pm$ 1.86	9.69 $\pm$ 1.89	9.25 $\pm$ 2.01	9.51 $\pm$ 2.17
Norameri	8.22 $\pm$ 3.88	11.32 $\pm$ 3.77	10.08 $\pm$ 4.03	11.16 $\pm$ 4.40	12.11 $\pm$ 6.07	10.55 $\pm$ 4.24	13.40 $\pm$ 7.81
Sinotibe	13.72 $\pm$ 4.91	15.85 $\pm$ 4.93	14.88 $\pm$ 4.97	15.22 $\pm$ 5.03	15.85 $\pm$ 5.30	15.50 $\pm$ 5.34	16.97 $\pm$ 5.90
Misc	23.05 $\pm$ 2.46	24.54 $\pm$ 2.53	24.56 $\pm$ 2.54	24.82 $\pm$ 2.52	25.13 $\pm$ 2.50	25.68 $\pm$ 2.55	27.47 $\pm$ 2.57
Average	10.80 $\pm$ 3.03	12.78 $\pm$ 2.90	12.25 $\pm$ 3.12	12.47 $\pm$ 3.10	12.82 $\pm$ 3.32	12.50 $\pm$ 3.22	13.44 $\pm$ 3.51

Table 16 Performance (CER) across language groupings for different upsampling conditions. CER is averaged across all languages within each language family; error bars indicate 95% Confidence Intervals.

Condition	High	Med	Low	Avg
cbeta_0.0_lbeta_0.0	6.28	6.12	21.14	11.18
cbeta_0.25_lbeta_0.5	6.80	8.70	23.23	12.91
cbeta_0.5_lbeta_0.5	6.54	8.08	23.42	12.5
cbeta_0.75_lbeta_0.5	6.60	7.63	24.38	12.68
cbeta_0.5_lbeta_0.25	6.54	7.86	23.09	12.89
cbeta_0.5_lbeta_0.75	6.50	8.37	23.79	12.87
cbeta_1.0_lbeta_1.0	6.75	8.09	26.40	13.75

Table 17 Performance (CER) across resource buckets and conditions for different *beta* values. The rightmost column (Avg) is separated for clarity.

Training Data	Holdout Data source	Holdout CER	Baseline CER	Baseline CERR
AllASR_x_mls	mls	4.3	3.27	-31%
AllASR_x_fleurs	fleurs	20.95	11.95	-75%
AllASR_x_cv22	cv22	33.46	19.57	-71%
MMS-lab	mls	6.34	3.27	-94%
MMS-lab	fleurs	35.72	11.95	-199%
MMS-lab	cv22	43.35	19.57	-1.22

Table 18 Corpus holdout ablation results. Rows 1-3 contain performance for models trained on our AllASR dataset, with a single source held-out from training. CERs for heldout corpora are shown in third column, and can be compared to CER obtained by a baseline model trained on all data (including the holdout corpus, fourth column). Rows 4-6 show holdout performance of a model trained on just MMS-lab, which covered all languages in the holdout corpora but had less audio diversity. Column 5 shows relative Character Error Rate reduction (CERR) of the holdout condition relative to the baseline: $(CER_{baseline} - CER_{treatment})/CER_{baseline}$. These values are all negative, indicating regressions for the holdout models compared to the baseline, which has seen all the data.

Babel, MLS, and CV22⁸. For example, *AllASR_x_mls* refers to a model trained on all of the above except MLS. We evaluate these *AllASR_x* holdout models on development splits from the held-out data sources, and compares them to a baseline model trained on the complete *AllASR* dataset, thus measuring their ability to generalize to unseen audio distributions.

Further, we contrasted these hold-out model conditions with a model trained on just MMS-lab. This latter model was still exposed to all languages in the hold-out sources, but it was not exposed audio from any other data sources. Comparing the *AllASR_x* holdout models against MMS-lab model allows us to assess the degree to which our model becomes better at generalizing to new audio distributions as we expand the training set to include more sources. In all conditions, we trained 1B CTC models for 100K steps at 32 GPUs.

Results are displayed in Table 18. Rows 1-3 show CERs obtained by *AllASR_x* models on their respective holdout corpora (column 3). These numbers can be compared against Baseline CERs obtained by the *AllASR* model (column 4). Baseline CERR (column 5) makes this delta explicit: more negative values indicate larger regressions compared to baseline. As expected, performance regresses for all held-out data sources compared to the baseline. The regression is more pronounced on FLEURS and CV22 than on MLS, suggesting that those two sources comprise more distinct audio distributions compared to the other sources within AllASR. That said, models still perform reasonably well on the holdout corpora (especially on MLS), indicating an ability to generalize to unseen audio distributions.

Crucially, Baseline CERR is substantially better in the *AllASR_x* models compared to the MMS-lab condition. This is true across all three holdout sources and indicates that our *AllASR* recipe improves our model’s ability to generalize to unseen audio distributions, as compared to training on a single data source with the same language coverage.

5.7.3 Model Robustness to Background Noise

Building on the previous section, we further examine model robustness by measuring ASR performance as a function of background noise and/or clarity of the speech signals. To do this, we ran audio samples in our development sets through the Torchaudio Squim models, which emit estimations of speech audio quality (Kumar et al., 2023). Figure 8 shows CER as a function of SI-SDR, which is a model estimate of the level of background noise relative to the speech signal. Model performance on different language groups is shown in different colors according to the resource-level (number of training hours) associated with each language. The analysis was performed on our 7B CTC model (solid line) as well as our 7B LLM-ASR model (dashed line).

Results are presented in Figure 8. Each utterance was binned into SI-SDR ranges, which were not evenly spaced but instead selected to showcase the extreme outliers in the distribution of our AllASR dataset (i.e., audios with large amounts of background noise). The ranges correspond to the following SI-SDR percentiles: [0-1, 1-5, 5-20, 20-40, 40-60, 60-80, 80-95, 95-100]. To remove any confounds with LID, we only include

⁸Note this is a subset of AllASR used to train our final models. Refer to Table 3 for a list of all data sources

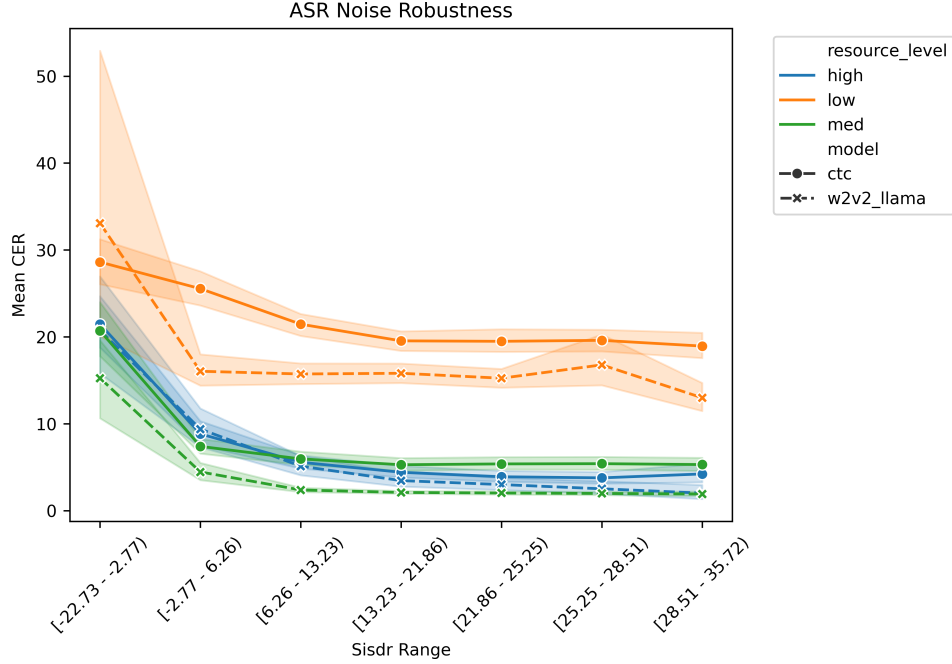


Figure 8 ASR noise robustness across All_ASR dev sets. Utterances were binned into SI-SDR ranges that showcase the outliers with low SI-SDR values, up through the rest of the distribution. Mean CER values, averaged across languages (y-axis), are plotted against SI-SDR range (x-axis) of the associated audio. Error ribbons indicate 95% CI. Results are further grouped by resource-level of the included language, as indicated by color: low- (orange; <10 hrs), medium- (blue; $10 \leq \text{hrs} < 50$ hours), and high-resource (green; >50 hours). Results presented are for the 7B CTC model (solid line) and w2v2_LLM (dashed line). Error ribbons indicate 95% Confidence Intervals.

languages with utterances in each of the displayed SI-SDR bins. Within each SI-SDR bin, we obtain Mean CER (averaged across languages; y-axis) and plot this against SI-SDR bin-range (x-axis). Error ribbons indicate 95% Confidence Intervals. CTC performance is indicated by dots/solid line, while LLM-ASR is indicated by x/dashed line. Languages with different levels of training data are grouped by color: low-resource (<10 hours), medium-resource (between 10-50 hours), and high-resource (>50 hours).

As expected, CER is higher for utterances with low SI-SDR values (high background noise) compared to utterances with higher SI-SDR (cleaner audio). CER is highest and most variable at the extreme low-end (lowest 1% of SI-SDR). However, CER quickly drops and flattens out after this. For instance, even for the noisiest 1%-5% of utterances, LLM-ASR model obtains CERs ≤ 10 across all language groups, and the CTC model obtains CERs < 15 for medium- and high-resource languages. In the remaining SI-SDR bins, CER is quite flat within each language group. It is important to recall that the x-axis in Figure 8 is not a linear scale throughout: the first two bin-ranges represent outlier utterances with extreme levels of background noise (i.e., top 1% and top 5%, respectively). Overall, these results indicate good model robustness to moderate levels of background noise (i.e., lowest 5% percentiles), and that our models do not exhibit any bias in background noise sensitivity as a function of language resource-level.

5.7.4 Omnilingual + OMSF ASR Holdout Ablation

To measure the value of the Omnilingual + OMSF ASR data (i.e., all the new data collected in this project: Omnilingual ASR Corpus plus OMSF), we ran a simple ablation in which we compared a ALLASR model against an ALLASR_x_Omnilingual + OMSF ASR model. In the latter, we held out Omnilingual + OMSF ASR data from training and then evaluated the model on the hold-out Omnilingual + OMSF ASR dev sets. In both conditions, we trained 7B LLM-ASR models for 150K steps across 64 GPUs.

To be clear, Omnilingual + OMSF ASR introduces mostly new languages to the mix, so in these cases, the ALLASR_x_Omnilingual + OMSF ASR is being evaluated on languages it was not exposed to during

Data Condition	CER (new languages)	CER (overlap langs)
<i>AllASR_x_OMNI</i>	47.03	39.46
<i>AllASR_OMNI</i>	22.62	11.50

Table 19 Omnilingual + OMSF ASR holdout ablation results. Mean CERs on the Omnilingual + OMSF ASR dev sets, averaged across languages are shown for the holdout and full-data conditions. Results reported separately for new languages introduced by Omnilingual + OMSF ASR versus languages that were already present in other corpora within ALLASR.

training. In these cases, we expect the ALLASR model to outperform the holdout model. Nevertheless, we include the ablation results to validate the training signal in Omnilingual + OMSF ASR; this allows us to ensure that our claims of supporting newly introduced languages are well founded.

Additionally, there are 13 overlapping languages in Omnilingual + OMSF ASR that are also contained in other corpora within ALLASR. For these languages, we would like to see if the additional Omnilingual + OMSF ASR training data provides a valuable signal above and beyond what was already present in our training data, especially with regard to speaker diversity and more naturalistic audio conditions. We separately report ablation results for new and overlapping languages in Table 19.

Results in Table 19 highlight the value of the Omnilingual + OMSF ASR data collected in this project, both by extending coverage to new languages and by substantially improving performance on already-supported ones. For new languages, our AllASR_OMNI model achieves a mean CER of 22.62, less than half the 47.03 obtained by the holdout model. Although 22.62 remains relatively high compared to CERs obtained on other corpora, it nevertheless represents a major reduction from the holdout model’s zero-shot performance, despite that model being highly multilingual. For overlapping languages, the impact of Omnilingual + OMSF ASR data is even more striking: CERs drop from 39.46 with the holdout model to 11.50 with AllASR_OMNI.

This latter result underscores the fact that data from Omnilingual + OMSF ASR is quite challenging for ASR compared to many pre-existing multilingual datasets, which mostly consist of clean, studio-quality recordings of speaker-reading. Omnilingual + OMSF ASR was intentionally curated to represent naturalistic (i.e., often noisy) audio conditions, diverse speaker identities, and spontaneous, expressive speech. The benefits of such data are demonstrated here: without including them in the datamix, an equally multilingual model (i.e., our holdout) struggles in these more difficult, but more naturalistic audio/speaker conditions. In sum, by including Omnilingual + OMSF ASR, we introduce new language coverage and also substantially improve model robustness, which ultimately situates our models for use in the wild.

5.7.5 Fine-tuning for Individual Low-Resource Languages

In this study, we fine-tuned bespoke CTC models on individual low-resource languages. There are two motivations here. First, from a theoretical standpoint, we are interested in establishing the best performance achievable for languages with fewer than 10 hours of data, and in quantifying the performance gap relative to our Omnilingual ASR models trained across 1,600+ languages. Second, we present our learnings to the community to provide recommended settings for users interested in adapting and optimizing our open-source models for their own bespoke purposes, especially in lower compute settings. This study was performed with 11 low-resource languages, with between 5-10 hours of training data and at least 1 hour of validation splits. See Table 20 for the complete list.

We fine-tuned language-specific CTC models for each of these 11 languages, across the 300M, 1B, and 3B scales. In one condition, we seeded from a pretrained w2v2 checkpoint, and in another, we seeded from an OmniASR CTC checkpoint, which was pretrained on all 1600+ languages. For the w2v2-seed condition, we trained with a learning rate of 1e-05 for 30K steps, though we observed that models typically converge within 10K steps. For the CTC-seed condition, we also use an lr of 1e-05 and trained for 5K steps. CTC fine-tuning takes 1 hour of walltime on 32 GPUs for the 300M scale. These hyperparameters were selected based on empirical sweeps for a couple of exemplar languages, but of course, in practice, the optimal training hyperparameters will be a function of the specific language and data used in finetuning. For example, we

LID	Script	# train hours	Best CER
ast	Latn	8.1	3.31
ckb	Arab	9.1	4.47
ltz	Latn	8.5	7.42
hsb	Latn	9.1	2.17
afo	Latn	7.6	29.7
ahl	Latn	7.2	16.47
div	Thaa	7.6	5.16
fuv	Latn	6.5	15.1
qxp	Latn	9.9	1.61
ajg	Latn	9.5	8.05
vro	Latn	9.5	5.7

Table 20 Low-resource languages used in language-specific study.

observed that certain languages converged long before the # training steps listed here.

We then compare the performance (CER) of these language-specific models against our Omnilingual ASR CTC models at each scale. These Omnilingual ASR models were trained on all 1600+ languages, without any sort of language-specific optimization. Results can be found in [Table 21](#). Language-specific models substantially outperform the Omnilingual ASR baselines, achieving CERs of less than 5 in many of these low-resource languages—even at the smaller 300M and 1B scales. Additionally, CTC-seeded models consistently outperformed w2v2-seeded models at the 300M and 1B scales, even though they were fine-tuned for a fraction of the training steps (5K instead of 30K). Consequently, we advise practitioners wishing to optimize our 300M and 1B models for ASR in particular low-resource languages to seed with CTC checkpoints. However, at the 3B scale the w2v2-seeded checkpoints trained for 30K steps generally outperformed the ctc-seeded checkpoints trained for 5K steps.

[Table 21](#) also shows CERs obtained by our 7B OmniASR LLM model in the rightmost column. In most cases, the OmniASR 7B-LLM was quite competitive with the language-specific models, indicating an extremely high performance on these low-resource languages despite the fact that it was trained on all 1600+ languages and without any language-specific optimization. On the other hand, even though the language-specific models are significantly smaller than the 7B-LLM model and lack the LLM architectural component, they still obtained lower CERs for most languages, even at the smallest 300M scale. This demonstrates a unique strength of our open-source Omnilingual ASR models: they contain rich omnilingual knowledge, and can be quickly adapted and fine-tuned to excel in particular low-resource settings with minimal compute. Once fine-tuned, the lightweight CTC models can be run in small compute environments during inference, which can be desirable in numerous applications.

5.8 Impact of Conditioning on Language Codes

We performed an ablation experiment to study the impact of conditioning the model on the ID of the language and script combination as described in [Section 4.5](#). Models trained with this feature can be evaluated with or without providing the language and script information. To measure its effect, we compared a model trained without language and script ID conditioning against models trained with different probabilities of including this information during training.

The results in [Table 22](#) show that compared to a baseline trained without conditioning, training with language and script conditioning on at least 50% of the samples yields considerable improvements on FLEURS-102 and Common Voice when conditioning is used at inference. These accuracy gains largely come from utterances that, without conditioning, were misrecognized in the wrong language or script—errors that significantly increased CER. Importantly, training with conditioning applied to only half of the batches preserved the model’s ability to operate effectively without conditioning at inference, still recognizing the correct language and script for the vast majority of samples. In fact, this setup showed virtually no degradation in accuracy compared to the baseline model (training language conditioning for 0% of the samples) when conditioning

Language	Scale	Single-Lang		OmniCTC	OmniLLM (7B)
		CTC Seed	W2V2 Seed		
afo_Latn	300m	32.54	32.32	33.54	38.91
	1b	31.58	29.71	33.17	
	3b	30.89	29.11	32.18	
ahl_Latn	300m	18.78	20.52	44.28	24.33
	1b	17.66	16.47	36.76	
	3b	17.87	15.27	34.61	
ajg_Latn	300m	8.05	8.63	21.97	7.54
	1b	8.82	8.11	19.14	
	3b	9.02	7.92	15.63	
ast_Latn	300m	4.95	8.02	10.87	5.105
	1b	3.55	4.83	7.88	
	3b	3.91	3.31	6.44	
ckb_Arab	300m	5.82	8.01	15.29	4.73
	1b	5.05	5.91	12.28	
	3b	5.20	4.17	9.94	
div_Thaa	300m	5.54	8.36	19.21	5.58
	1b	5.16	5.66	17.21	
	3b	5.45	4.57	13.04	
fuv_Latn	300m	16.41	18.45	23.69	26.83
	1b	15.59	15.10	20.47	
	3b	15.14	14.35	16.31	
hsb_Latn	300m	2.93	7.18	10.41	4.1
	1b	2.57	2.17	7.07	
	3b	3.20	1.79	4.94	
ltz_Latn	300m	9.88	15.94	19.72	6.07
	1b	7.42	10.72	12.44	
	3b	8.09	7.12	9.80	
qxp_Latn	300m	1.70	2.08	4.49	1.32
	1b	1.61	1.68	2.94	
	3b	1.81	1.47	2.71	
vro_Latn	300m	7.18	9.39	16.74	4.02
	1b	6.36	5.70	12.67	
	3b	6.76	5.12	10.16	

Table 21 Model performance (CER) across low-resource languages and scales. Columns 3-4 show language-specific models. The rightmost column (OmniLLM (7B)) is separated for clarity.

Language Conditioning	Conditioning at Inference	MMS-Lab	Omnilingual ASR	Babel	FLEURS-102	MLS	CV22
0.0	No	2.5	13.3	19.1	7.9	2.6	11.3
0.2	No	2.5	13.4	19.3	7.4	2.6	11.8
	Yes	2.5	13.2	19.2	7.6	2.6	8.2
0.5	No	2.5	13.7	19.4	7.5	2.6	11.8
	Yes	2.5	13.4	19.3	7.1	2.6	7.9
1.0	No	15.7	42.5	54.1	34.7	3.1	45.1
	Yes	2.5	14.0	19.2	6.9	2.6	6.9

Table 22 Impact of language and script conditioning on the LLM-ASR model. A model with language and script conditioning 50% of the time during training is able to deliver best tradeoff between inference modes—when language and script information are either absent or provided.

was not applied at inference. Based on these findings, we adopt language and script conditioning for 50% of the samples during training in our final LLM-ASR models.

5.9 Comparison of OmniASR-W2V Models to Existing SSL Speech Encoders

In this section, we compare the OmniASR-W2V family with some of the most widely used multilingual SSL speech encoders, including XLSR- $\{0.3B, 1B, 2B\}$ from [Babu et al. \(2021\)](#) and MMS- $\{0.3B, 1B\}$ from [Pratap et al. \(2024\)](#). In [Table 23](#), we highlight the key differences among the models, focusing on the number of languages covered, the volume of pre-training data, and the model size measured in parameters.

Model	# of lang	Datasets	Data volume (hrs)	# of params
<i>Prior Work</i>				
XLSR-0.3B	128	VP, MLS, CV6, VL, BBL	436k	317M
XLSR-1B	128	VP, MLS, CV6, VL, BBL	436k	965M
XLSR-2B	128	VP, MLS, CV6, VL, BBL	436k	2162M
MMS-0.3B	1406	VP, MLS, CV9, VL, BBL, MMS-Lab, FL	491k	317M
MMS-1B	1406	VP, MLS, CV9, VL, BBL, MMS-Lab, FL	491k	965M
<i>This Work</i>				
OmniASR-W2V-0.3B	1600+	SSLCORPUS (Section 3.3.4)	4.3M	317M
OmniASR-W2V-1B	1600+	SSLCORPUS	4.3M	965M
OmniASR-W2V-3B	1600+	SSLCORPUS	4.3M	3046M
OmniASR-W2V-7B	1600+	SSLCORPUS	4.3M	6488M

Table 23 Existing SSL speech encoders. VP, MLS, CV, VL, BBL, and FL stand for VoxPopuli, Multilingual LibriSpeech, Common Voice, VoxLingua, Babel, and FLEURS, respectively. Note that XLSR and MMS models used different versions of CV: CV6 and CV9, where the latter covers 29 more languages.

To enable a fair comparison, all pre-trained speech encoders were fine-tuned with CTC on ALLASR following the setting specified in [Section 5.1](#). We report the test set results on MMS-Lab, Omnilingual ASR Corpus, FLEURS-102, MLS, and CV22 in [Table 24](#).

Comparing models of the same size, we see that OmniASR-W2V-0.3B outperforms XLSR-0.3B and MMS-0.3B on all benchmarks except for MLS, where OmniASR-W2V-0.3B’s performance is on par with MMS-0.3B but worse than XLSR-0.3B. Note that while XLSR-0.3B outperforms OmniASR-W2V-0.3B by less than 10% on MLS, its performance on the rest of the benchmarks lags behind OminASR-W2V-0.3B by 18%, 42%, 16%, and 13%, respectively. A similar conclusion can be drawn from the comparison of OmniASR-W2V-1B, XLSR-1B, and MMS-1B, except for the fact that, now, OmniASR-W2V-1B beats MMS-1B in all cases, and the performance gap with XLSR-1B on MLS is reduced to 6%.

Model	MLS	FLEURS-102	MMS-Lab	CV22	Omnilingual ASR Corpus
<i>Prior Work</i>					
XLSR-0.3B	3.7	14.6	12.6	24.0	30.3
XLSR-1B	2.9	10.2	7.6	18.8	25.7
XLSR-2B	3.0	9.9	5.8	19.5	24.5
MMS-0.3B	4.1	14.2	8.2	22.2	29.1
MMS-1B	3.2	10.2	4.7	16.8	25.2
<i>This Work</i>					
OmniASR-W2V-0.3B	4.1	12.0	7.3	20.2	26.4
OmniASR-W2V-1B	3.1	8.9	4.5	16.5	24.1
OmniASR-W2V-3B	2.7	8.0	3.5	16.2	22.8
OmniASR-W2V-7B	2.5	7.5	3.1	15.8	20.8

Table 24 Results of existing SSL speech encoders and the OmniASR-W2V models. For each benchmark, we report the average CER across languages on the test set.

Scaling beyond 1B, we see OmniASR-W2V-3B and OmniASR-W2V-7B continue to widen the gap with other encoders across all benchmarks, suggesting they are the best choices for optimal performance on both top languages and long-tailed languages.

6 Societal Impact and Conclusion

Omnilingual ASR illustrates how scaling methods, when combined with deliberate data collection and new architectural innovation, can reshape the trajectory of multilingual ASR. The project not only extends coverage to more than 1,600 languages, with over 500 represented for the first time in any ASR system, but also reframes how coverage itself is conceived. In contrast to prominent existing systems (Radford et al., 2023; Pratap et al., 2024; Zhang et al., 2023), where unsupported languages could only be added through expert-driven fine-tuning, Omnilingual ASR demonstrates that recognition can be extended to entirely new languages with just a few in-context samples. This shift from fixed coverage to open-ended extensibility enables certain underserved groups to bring their languages into conversation with digital tools that have historically excluded them.

The coexistence of massive, high-accuracy models with lightweight 300M-parameter variants also alters the economics of deployment, making it feasible to adapt ASR both to high-compute cloud infrastructures and to low-power devices in areas with limited connectivity. This flexibility broadens not only the range of research questions that can be pursued but also the contexts in which ASR can be applied, from speech-to-text translation pipelines to community-led archives. By open-sourcing models and training pipelines, Omnilingual ASR lowers the barriers to entry, shifting long-tail ASR research from a niche pursuit to a tractable and collaborative enterprise.

For language communities, the impact is both promising and contingent. Already, Omnilingual ASR is being deployed in practice: health practitioners in Nigeria are using the system to facilitate Hausa transcriptions in community clinics, with the intention of improving documentation and patient care. In oral cultures, it could help make endangered archives more searchable; in education, lightweight models might power interactive learning tools in mother tongues; in civic life, transcription of local-language broadcasts could expand access to news and information. Yet these same capabilities can also be repurposed in ways that conflict with community priorities, from surveillance to unwanted moderation (Abdullah et al., 2021). This tension underscores the need for participatory governance and ongoing dialogue, rather than one-time transfers of technology (Wang et al., 2024b).

Importantly, our community partners remind us of the need for large technology companies not only to draw on open language data but also to reinvest in its creation and stewardship. Omnilingual ASR was designed in this spirit: not as an act of charity, but as part of a healthy, respectful, and mutually beneficial ecosystem in which communities are compensated for the time and emotional labor that language documentation entails. In

light of ongoing discussions about consent and compensation in AI training data, it is essential to acknowledge that these concerns highlight the complexities surrounding ethical practices in this field of research. They point to longstanding issues of power, participation, and equity in how language resources are built and shared. Our approach—compensating native speakers and working through local partnerships—was one attempt to respond to these challenges. Still, compensation should not be seen as a panacea: some communities may prefer voluntary, crowdsourced participation, while others may feel financially pressured into contributing data. Although we did not observe such dynamics in our own experience, they remain a possibility and highlight the importance of vigilance in future work to ensure that participation is informed, voluntary, and aligned with community priorities.

Reflecting on the project’s trajectory, several broader lessons emerge. First, the long tail of languages should not be treated as a final frontier to be “solved” once and for all, but as a dynamic, evolving space of collaboration in which linguistic, technical, and social knowledge interact. Second, open-sourcing at this scale is not merely an act of transparency but an intervention that redistributes the power to innovate, enabling actors historically excluded from large-scale AI development. Third, large-scale ASR is inseparable from the politics of data: how it is gathered, who is compensated, and who retains influence over its use (Reitmaier et al., 2022).

Looking ahead, Omnilingual ASR can serve as a foundation for broader research agendas that connect ASR to multimodal AI, language preservation, and participatory technology governance. Future directions include combining Omnilingual ASR with large language models to support conversational agents in under-resourced languages, embedding it in community-run archives to keep linguistic data locally controlled, and expanding its role in speech translation technologies. At the same time, sustaining open multilingual resources at this scale will require policymakers, funders, and interdisciplinary researchers to confront how to share responsibility for building and maintaining them in ways that prioritize long-term community needs (Wang et al., 2024b). By situating innovation within these broader ethical and institutional contexts, Omnilingual ASR seeks not only to advance the state-of-the-art but also to reshape the terms of engagement for how the next generation of community-focused AI will be built, shared, and governed.

References

- Solomon Teferra Abate, Wolfgang Menzel, and Bairu Tafila. An amharic speech corpus for large vocabulary continuous speech recognition. In *INTERSPEECH-2005*, 2005.
- Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems. In *2021 IEEE symposium on security and privacy (SP)*, pages 730–747. IEEE, 2021.
- Basil Abraham, Danish Goel, Divya Siddarth, Kalika Bali, Manu Chopra, Monojit Choudhury, Pratik Joshi, Preethi Jyoti, Sunayana Sitaram, and Vivek Seshadri. Crowdsourcing speech data for low-resource languages from low-income workers. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2819–2826, 2020.
- Gilles Adda, Sebastian Stüker, Martine Adda-Decker, Odette Ambourou, Laurent Besacier, David Blachon, Hélène Bonneau-Maynard, Pierre Godard, Fatima Hamlaoui, Dmitry Idiatov, et al. Breaking the unwritten language barrier: The bulb project. *Procedia Computer Science*, 81:8–14, 2016.
- Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. MEGA: Multilingual evaluation of generative AI. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.258. <https://aclanthology.org/2023.emnlp-main.258/>.
- Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, 2020.
- Akari Asai, Sneha Kudugunta, Xinyan Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. BUFFET: Benchmarking large language models for few-shot cross-lingual transfer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.
- Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick Von Platen, Yatharth Saraf, Juan Pino, et al. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*, 2021.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in neural information processing systems*, 2020.
- Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Unsupervised speech recognition. *Advances in Neural Information Processing Systems*, 34:27826–27839, 2021.
- Can Balioglu, Martin Gleize, Artyom Kozhevnikov, Ilia Kulikov, Tuan Tran, and Julien Yao. fairseq2, 2023. <http://github.com/facebookresearch/fairseq2>.
- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.45. <https://aclanthology.org/2023.ijcnlp-main.45/>.
- Ankur Bapna, Colin Cherry, Yu Zhang, Ye Jia, Melvin Johnson, Yong Cheng, Simran Khanuja, Jason Riesa, and Alexis Conneau. mslam: Massively multilingual joint pre-training for speech and text. *arXiv preprint arXiv:2202.01374*, 2022.
- Martijn Bartelds, Nay San, Bradley McDonnell, Dan Jurafsky, and Martijn Wiering. Making more of little data: Improving low-resource automatic speech recognition using data augmentation. *arXiv preprint arXiv:2305.10951*, 2023.
- Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech communication*, 56:85–100, 2014.

- Steven Bird. Must nlp be extractive? In *62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pages 14915–14929. Association for Computational Linguistics (ACL), 2024.
- Alan W Black. Cmu wilderness multilingual speech dataset. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5971–5975. IEEE, 2019.
- Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. Systematic inequalities in language technology performance across the world’s languages. *arXiv preprint arXiv:2110.06733*, 2021.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022a.
- William Chen, Wangyou Zhang, Yifan Peng, Xinjian Li, Jinchuan Tian, Jiatong Shi, Xuankai Chang, Soumi Maiti, Karen Livescu, and Shinji Watanabe. Towards robust speech representation learning for thousands of languages, 2024. <https://arxiv.org/abs/2407.00837>.
- Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro Moreno, Ankur Bapna, and Heiga Zen. Maestro: Matched speech text representations through modality matching. *arXiv preprint arXiv:2204.03409*, 2022b.
- Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, 2022.
- Anna Seo Gyeong Choi and Hoon Choi. Fairness of automatic speech recognition: Looking through a philosophical lens. *arXiv preprint arXiv:2508.07143*, 2025.
- Kwanghee Choi, Ankita Pasad, Tomohiko Nakamura, Satoru Fukayama, Karen Livescu, and Shinji Watanabe. Self-supervised speech representations are more phonetic than semantic. *Interspeech*, 2024.
- Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28, 2015.
- Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250, 2021.
- Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE, 2023.
- Ned Cooper, Courtney Heldreth, and Ben Hutchinson. "it's how you do things that matters": Attending to process to better serve indigenous communities with language technologies. *arXiv preprint arXiv:2402.02639*, 2024.
- Ken H Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.
- Digital Umuganda. Afrivoice_Ethopia_Amharic v1, July 2025a. <https://doi.org/10.5281/zenodo.16569778>.
- Digital Umuganda. Afrivoice_Ethopia_Afaan_Oromo v1, July 2025b. <https://doi.org/10.5281/zenodo.16563198>.
- Digital Umuganda. Afrivoice_Ethopia_Sidama v1, July 2025c. <https://doi.org/10.5281/zenodo.16574482>.
- Digital Umuganda. Afrivoice_Ethopia_Tigrinya v1, July 2025d. <https://doi.org/10.5281/zenodo.16575590>.
- Digital Umuganda. Afrivoice_Ethopia_Wolaytta v1, July 2025e. <https://doi.org/10.5281/zenodo.16576405>.
- Digital Umuganda. Afrivoice kinyarwanda, 2025f. https://huggingface.co/datasets/DigitalUmuganda/Afrivoice_Kinyarwanda.
- Digital Umuganda. Afrivoice swahili, 2025g. https://huggingface.co/datasets/DigitalUmuganda/Afrivoice_Swahili.
- Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. Sonar: sentence-level multimodal and language-agnostic representations. *arXiv preprint arXiv:2308.11466*, 2023.
- Chris Emezue, NaijaVoices Community, Busayo Awobade, Abraham Owodunni, Handel Emezue, Gloria Monica To-bechukwu Emezue, Nefertiti Nneoma Emezue, Sewade Ogun, Bunmi Akinremi, David Ifeoluwa Adelani, et al. The naijavoices dataset: Cultivating large-scale, high-quality, culturally-rich speech data for african languages. *arXiv preprint arXiv:2505.20564*, 2025.

- Joshua A Fishman. *Reversing language shift: Theoretical and empirical foundations of assistance to threatened languages*, volume 76. Multilingual matters, 1991.
- Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. Segment any text: A universal approach for robust, efficient and adaptable sentence segmentation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11908–11941, Miami, Florida, USA, November 2024. Association for Computational Linguistics. <https://aclanthology.org/2024.emnlp-main.665>.
- M. J. F. Gales, K. M. Knill, A. Ragni, and S. P. Rath. Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED. In *Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2014)*, 2014.
- John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, and David S. Pallett. Timit acoustic-phonetic continuous speech corpus. Technical Report LDC93S1, Linguistic Data Consortium, Philadelphia, 1993. <https://catalog.ldc.upenn.edu/LDC93S1>.
- Elodie Gauthier, Laurent Besacier, Sylvie Voisin, Michael Melese, and Uriel Pascal Elingui. Collecting resources in sub-saharan african languages for automatic speech recognition: a case study of wolof. *LREC*, 2016.
- Hadrien Gelas, Laurent Besacier, and Francois Pellegrino. Developments of Swahili resources for an automatic speech recognition system. In *SLTU - Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2012.
- Rafael Mosquera Gómez, Julián Eusse, Juan Ciro, Daniel Galvez, Ryan Hileman, Kurt Bollacker, and David Kanter. Speech wikimedia: A 77 language multilingual speech dataset. *arXiv preprint arXiv:2308.15710*, 2023.
- Jeff Good and Calvin Hendryx-Parker. Modeling contested categorization in linguistic databases. In *Proceedings of the EMELD 2006 Workshop on Digital Language Documentation: Tools and standards: The state of the art*, 2006.
- Deepa P Gopinath, Thennal D K, Vrinda V Nair, Swaraj K S, and Sachin G. IMA SC – ICFOSS malayalam speech corpus. *arXiv preprint arXiv:2211.12796*, 2022. <https://arxiv.org/abs/2211.12796>.
- Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR, 2014.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *International conference on Machine learning*, 2006.
- Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.
- Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. Glottolog 5.1., 2024. <https://glottolog.org/>.
- Fei He, Shan-Hui Cathy Chu, Oddur Kjartansson, Clara Rivera, Anna Katanova, Alexander Gutkin, Isin Demirsahin, Cibu Johny, Martin Jansche, Supheakmungkol Sarin, and Knot Pipatsrisawat. Open-source Multi-speaker Speech Corpora for Building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech Synthesis Systems. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, 2020.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, 2019.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.
- Kedir Yassin Hussen, Waleign Tewabe Sewunetie, Abinew Ali Ayele, Sukairaj Hafiz Imam, Shamsuddeen Hassan Muhammad, and Seid Muhie Yimam. The state of large language models for african languages: Progress and challenges. *arXiv preprint arXiv:2506.02280*, 2025.
- Sukairaj Hafiz Imam, Babangida Sani, Dawit Ketema Gete, Bedru Yimam Ahamed, Ibrahim Said Ahmad, Idris Abdulmumin, Seid Muhie Yimam, Muhammad Yahuza Bello, and Shamsuddeen Hassan Muhammad. Automatic speech recognition for african low-resource languages: Challenges and future directions. *arXiv preprint arXiv:2505.11690*, 2025.

- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*, 2020.
- KenCorpus Consortium. African next voices: Pilot data collection in kenya, 2025.
- Yerbolat Khassanov, Saida Mussakhjayeva, Almas Mirzakhmetov, Alen Adiyev, Mukhamet Nurpeiissov, and Huseyin Atakan Varol. A crowdsourced open-source Kazakh speech corpus and initial speech recognition baseline. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2018.
- Nithin Rao Koluguri, Monica Sekoyan, George Zelenfroynd, Sasha Meister, Shuoyang Ding, Sofia Kostandian, He Huang, Nikolay Karpov, Jagadeesh Balam, Vitaly Lavrukhin, et al. Granary: Speech recognition and translation dataset in 25 european languages. *arXiv preprint arXiv:2505.13404*, 2025.
- András Kornai. Digital language death. *PloS one*, 8(10):e77056, 2013.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan Van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72, 2022.
- Anurag Kumar, Ke Tan, Zhaoheng Ni, Pranay Manocha, Xiaohui Zhang, Ethan Henderson, and Buye Xu. Torchaudio-squim: Reference-less speech quality and intelligibility measures in torchaudio. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- Colin Leong, Joshua Nemecek, Jacob Mansdorfer, Anna Filighera, Abraham Owodunni, and Daniel Whitenack. Bloom library: Multimodal datasets in 300+ languages for a variety of downstream tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8608–8621, 2022.
- Song Li, Yongbin You, Xuezhi Wang, Zhengkun Tian, Ke Ding, and Guanglu Wan. Msr-86k: An evolving, multilingual corpus with 86,300 hours of transcribed audio for speech recognition research. In *Proc. Interspeech 2024*, pages 1245–1249, 2024.
- Xinjian Li, Florian Metze, David R Mortensen, Alan W Black, and Shinji Watanabe. Asr2k: Speech recognition for around 2000 languages without audio. *Interspeech 2022*, 2022.
- Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. Yodas: Youtube-oriented dataset for audio and speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023.
- Nikola Ljubešić, Danijel Koržinek, Peter Rupnik, and Ivo-Pavao Jazbec. ParlaSpeech-HR - a freely available ASR dataset for Croatian bootstrapped from the ParlaMint corpus. In *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*, 2022.
- Julia Mainzinger and Gina-Anne Levow. Fine-tuning ASR models for very low-resource languages: A study on mvskoke. In Xiyan Fu and Eve Fleisig, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 76–82, Bangkok, Thailand, August 2024. Association for Computational Linguistics. ISBN 979-8-89176-097-4. doi: 10.18653/v1/2024.acl-srw.16. <https://aclanthology.org/2024.acl-srw.16/>.
- Vukosi Marivate, Kayode Olaleye, Sitwala Mundia, Nia Zion Van Wyk, Andinda Bakainga, Unarine Netshifhefhe, Mahmooda Milanzie, Hope Tsholofelo Mogale, Chijioke Okorie, Graham Morrissey, Dale Dunbar, Tsosheletso Chidi, Rooweither Mabuya, Andiswa Bukula, Respect Mlambo, and Tebogo Macucwa. Swivuriso: Creating the south african next voices multilingual speech dataset, 2025.
- David Erik Mollberg, Ólafur Helgi Jónsson, Sunneva Þorsteinsdóttir, Steinþór Steingrímsson, Eydís Huld Magnúsdóttir, and Jon Guðnason. Samrómur: Crowd-sourcing data collection for Icelandic speech recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020.
- NLLB Team. Scaling neural machine translation to 200 languages. *Nature*, 630:841–846, June 2024. doi: 10.1038/s41586-024-07335-x. <https://www.nature.com/articles/s41586-024-07335-x>.

- Millicent Ochieng, Varun Gumma, Sunayana Sitaram, Jindong Wang, Vishrav Chaudhary, Keshet Ronen, Kalika Bali, and Jacki O'Neill. Beyond metrics: Evaluating LLMs effectiveness in culturally nuanced, low-resource real-world scenarios. In Constantine Lignos, Idris Abdulmumin, and David Adelani, editors, *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 230–247, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-257-2. doi: 10.18653/v1/2025.africanlp-1.33. <https://aclanthology.org/2025.africanlp-1.33/>.
- Yin May Oo, Theeraphol Wattanavekin, Chenfang Li, Pasindu De Silva, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, Oddur Kjartansson, and Alexander Gutkin. Burmese Speech Corpus, Finite-State Text Normalization and Pronunciation Grammars with an Application to Text-to-Speech. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, 2020.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.
- Kyubyong Park and Thomas Mulc. Cst10: A collection of single speaker speech datasets for 10 languages. *Interspeech*, 2019.
- Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *Interspeech 2020*, 2020.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52, 2024.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, 2023.
- Thomas Reitmaier, Electra Wallington, Dani Kalarikalayil Raju, Ondrej Klejch, Jennifer Pearson, Matt Jones, Peter Bell, and Simon Robinson. Opportunities and challenges of automatic speech recognition systems for low-resource language speakers. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–17, 2022.
- Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. <https://doi.org/10.1561/1500000019>.
- Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.
- SEAMLESS Communication Team. Joint speech and text machine translation for up to 100 languages. *Nature*, 637: 587–593, January 2025. doi: 10.1038/s41586-024-08359-z. <https://www.nature.com/articles/s41586-024-08359-z>.
- Keshan Sodimana, Knot Pipatsrisawat, Linne Ha, Martin Jansche, Oddur Kjartansson, Pasindu De Silva, and Supheakmungkol Sarin. A Step-by-Step Process for Building TTS Voices Using Open Source Data and Framework for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2018.
- Per Erik Solberg and Pablo Ortiz. The Norwegian parliamentary speech corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 2014.
- VAANI Team. Vaani: Capturing the language landscape for an inclusive digital india (phase 1). <https://vaani.iisc.ac.in/>, 2025.
- Duc Chung Tran. FPT Open Speech Dataset (FOSD) - Vietnamese, 2020. <https://data.mendeley.com/datasets/k9sxxg2twv4/4>.
- Jörgen Valk and Tanel Alumäe. VoxLingua107: a dataset for spoken language recognition. In *Proc. IEEE SLT Workshop*, 2021.
- Daniel van Niekirk, Charl van Heerden, Marelie Davel, Neil Kleynhans, Oddur Kjartansson, Martin Jansche, and Linne Ha. Rapid development of TTS corpora for four South African languages. In *Proc. Interspeech 2017*, 2017.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.
- Changhan Wang, Anne Wu, and Juan Pino. Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*, 2020.
- Changhan Wang, Morgane Rivière, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *ACL 2021-59th Annual Meeting of the Association for Computational Linguistics*, 2021.
- Siyin Wang, Chao-Han Yang, Ji Wu, and Chao Zhang. Can whisper perform speech-based in-context learning? In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13421–13425. IEEE, 2024a.
- Skyler Wang, Ned Cooper, and Margaret Eby. From human-centered to social-centered artificial intelligence: Assessing chatgpt’s impact through disruptive events. *Big Data & Society*, 11(4):20539517241290220, 2024b.
- Hemant Yadav and Sunayana Sitaram. A survey of multilingual models for automatic speech recognition. *arXiv preprint arXiv:2202.12576*, 2022.
- Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vassilina Nikoulina. BLOOM+1: Adding language support to BLOOM for zero-shot prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.
- Steve Young. A review of large-vocabulary continuous-speech. *IEEE signal processing magazine*, 13(5):45, 1996.
- Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. Google USM: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*, 2023.
- Jinming Zhao, Vineel Pratap, and Michael Auli. Scaling a simple approach to zero-shot speech recognition. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Appendix

A Omnilingual ASR Language Coverage

Code	Name	Res	Code	Name	Res	Code	Name	Res
aae_Latn	Arbëreshë Albanian	L	bcw_Latn	Bana	M	bzh_Latn	Mapos Buang	M
aal_Latn	Afade	L	bcy_Latn	Bacama	L	bzi_Thai	Bisu	M
abb_Latn	Bankon	L	bcz_Latn	Bainouk-Gunyaamolo	L	bzi_Latn	Belize Kriol English	M
abi_Latn	Abidji	M	bda_Latn	Bayot	L	bzw_Latn	Basa (Nigeria)	L
abk_Cyrl	Abkhazian	M	bde_Latn	Bade	L	caa_Latn	Chorti	M
abn_Latn	Abua	L	bdg_Latn	Bonggi	M	cab_Latn	Garifuna	M
abp_Latn	Abellen Ayta	M	bdi_Latn	Baka (South Sudan)	H	cac_Latn	Chuj	M
abr_Latn	Abron	L	bdm_Latn	Buduma	L	cak_Latn	Kaqchikel	H
abs_Latn	Ambonese Malay	L	bdq_Latn	Bahnar	M	cap_Latn	Chipaya	M
aca_Latn	Achagua	M	bdu_Latn	Oroko	M	car_Latn	Galibi Carib	M
acd_Latn	Gikydyode	M	beb_Latn	Bebele	L	cas_Latn	Tsimané	M
ace_Latn	Achinese	M	beb_Latn	Bialí	M	cat_Latn	Catalan	H
acf_Latn	Saint Lucian Creole French	M	bel_Cyrl	Belarusian	H	cax_Latn	Chiquitano	M
ach_Latn	Acoli	M	bem_Latn	Bemba (Zambia)	M	cbc_Latn	Carapana	M
acm_Arab	Mesopotamian Arabic	L	ben_Beng	Bengali	H	cbl_Latn	Chachi	H
acn_Latn	Achaang	M	bep_Latn	Besoa	M	cbr_Latn	Cashibo-Cacataibo	M
acr_Latn	Achi	M	bew_Latn	Betawi	M	cbs_Latn	Cashinahua	M
acu_Latn	Achuar-Shiwiar	M	bex_Latn	Jur Modo	M	cbt_Latn	Chayahuita	M
acw_Arab	Hijazi Arabic	M	bfa_Latn	Bari	M	cbu_Latn	Candoshi-Shapra	M
ade_Latn	Adole	M	bfd_Latn	Bafut	L	cbv_Latn	Cacua	M
adh_Latn	Adhola	M	bfo_Latn	Malba Birifor	M	ccs_Latn	Chopi	M
adj_Latn	Adioukrou	M	bft_Arab	Balti	M	ccg_Latn	Samba Daka	L
adz_Tibt	Amdo Tibetan	H	bfy_Deva	Bagheli	M	cco_Latn	Comaltepec Chinantec	M
ady_Cyrl	Adyghe	M	bfz_Deva	Mahasu Pahari	M	cdj_Deva	Churahi	M
aeb_Arab	Tunisian Arabic	M	bgc_Deva	Haryanvi	M	cds_Latn	Chin Dong Chinese	M
aec_Arab	Saidi Arabic	L	bgp_Arab	Eastern Balochi	L	ceb_Latn	Cebuano	L
aeu_Latn	Akeu	M	bgq_Deva	Bagri	M	ceg_Latn	Chamacoco	M
afb_Arab	Gulf Arabic	M	bgr_Latn	Bawm Chin	M	cek_Latn	Eastern Khumi Chin	M
afg_Latn	Eloyi	L	bgt_Latn	Bughotu	M	cen_Latn	Central Chinese	M
afz_Latn	Afrikaans	H	bgw_Deva	Bhatri	M	ces_Latn	Czech	L
agd_Latn	Agarabi	M	bha_Deva	Bharia	L	cfa_Latn	Dijim-Bwilim	L
agg_Latn	Angor	L	bhb_Deva	Bhili	L	cfm_Latn	Palam Chin	M
agu_Latn	Agutaynen	L	bhh_Cyrl	Bukharic	L	cgc_Latn	Kagayanen	M
agr_Latn	Aguaruna	M	bho_Deva	Bhojpuri	L	cgg_Latn	Chiga	M
agu_Latn	Aguacateco	M	bhp_Latn	Bima	L	che_Cyrl	Chechen	M
agx_Cyrl	Aghul	L	bht_Deva	Bhattiyali	M	chf_Latn	Tabasco Chontal	L
aha_Latn	Ahauna	M	bht_Latn	Bada (Indonesia)	M	chg_Latn	Quioitepec Chinantec	L
ahk_Latn	Akha	M	bib_Latn	Bissa	M	chv_Cyrl	Chuvash	M
ahl_Latn	Igo	L	bim_Latn	Bimoba	M	chz_Latn	Ozumacín Chinantec	M
ahs_Latn	Ashe	L	bis_Latn	Bislama	M	cjk_Latn	Chokwe	L
ala_Latn	Arosi	M	biv_Latn	Southern Birifor	M	cjo_Latn	Ashéninka Pajonal	H
alg_Latn	Aja (Benin)	L	bji_Deva	Kanauji	L	qjb_Arab	Shor	L
aka_Latn	Akan	M	bjk_Latn	Barok	L	cjs_Cyrl	Shor	L
akb_Latn	Batak Angkola	M	bjn_Latn	Banjar	L	ckb_Arab	Central Kurdish	M
ake_Latn	Akawaio	M	bjr_Latn	Binumarien	H	ckl_Latn	Cibak	L
akp_Latn	Akpa	M	bjt_Latn	Balanta-Ganja	L	cko_Latn	Anifo	L
ala_Latn	Alago	M	biv_Latn	Bedjond	M	ckr_Latn	Kairak	L
alj_Latn	Alangan	L	bji_Latn	Bakwé	M	ckt_Cyrl	Chukot	L
aln_Latn	Gheg Albanian	L	biz_Latn	Baruga	M	cky_Latn	Kafem-Mushere	M
alo_Latn	Laríke-Wakasihu	L	bkd_Latn	Binukid	M	cln_Latn	Ron	L
alp_Latn	Alune	M	bkh_Latn	Bakoko	M	clm_Latn	Lealao Chinantec	M
als_Latn	Tosk Albanian	M	bkm_Latn	Kom (Cameroon)	L	clt_Latn	Eastern Highland Chatino	M
alt_Cyrl	Southern Altai	M	bkv_Latn	Bekwarra	M	cme_Latn	Cerma	M
alz_Latn	Alur	M	bky_Latn	Bokyi	L	cmn_Hans	Mandarin Chinese	H
ame_Latn	Yanesha'	M	bli_Latn	Balanta-Kentohe	L	cmn_Hant	Mandarin Chinese	H
amf_Latn	Hamer-Banna	M	bih_Latn	Kuwaas	M	cmo_Khmr	Central M'nom	M
amh_Ethi	Amharic	H	bit_Latn	Tai Dam	M	cmo_Latn	Central M'nom	M
ami_Latn	Amis	M	bix_Latn	Mag-Indi Ayta	M	cmr_Latn	Mro-Khimi Chin	M
amk_Latn	Ambai	H	biz_Latn	Balintak	M	cnh_Latn	Hakha Chin	M
amu_Latn	Guerrero Amuzgo	L	bmn_Latn	Northern Betsimisarakala Malagasy	M	cnl_Latn	Ashéninka	M
anc_Latn	Ngas	L	bmj_Latn	Bomu	M	cnl_Latn	Lalana Chinantec	M
ank_Latn	Goemai	L	bmr_Latn	Muinane	H	cnt_Latn	Tepetitotla Chinantec	M
ann_Latn	Ohloho	M	bmu_Latn	Somba-Siawari	M	coe_Latn	Koreguaje	M
anp_Deva	Angika	L	bmv_Latn	Bum	M	col_Latn	Colordo	M
anw_Latn	Anaang	L	bng_Beng	Benga	M	cok_Latn	Santa Teresa Cora	H
any_Latn	Anyin	M	bnn_Latn	Batanga	M	con_Latn	Cofán	M
aom_Latn	Omie	L	bnn_Latn	Bunun	L	cor_Latn	Cornish	L
aoz_Latn	Uab Meto	L	bno_Latn	Bantoanon	H	cot_Latn	Caquinte	H
apb_Latn	Sa'a	M	bnp_Latn	Bola	M	cou_Latn	Wamey	M
apc_Arab	Levantine Arabic	L	bns_Deva	Bundeli	L	cpa_Latn	Palantla Chinantec	M
apd_Arab	Sudanese Arabic	L	boa_Latn	Bora	M	cpb_Latn	Ucayali-Yurúa Ashéninka	H
apr_Latn	Arop-Lokep	M	bod_Tibt	Tibetan	H	cpu_Latn	Pichis Ashéninka	M
arb_Arab	Standard Arabic	M	boj_Latn	Anjam	M	cpu_Latn	Pu-Xian Chinese	L
arg_Latn	Aragonese	H	bom_Latn	Berom	M	cpy_Latn	South Ucayali Ashéninka	L
ari_Latn	Arabela	H	bos_Latn	Boróro	H	crh_Cyrl	Crimean Tatar	M
arq_Arab	Algerian Arabic	M	bot_Latn	Bosnian	H	crk_Cans	Plains Cree	M
ars_Arab	Najdi Arabic	M	bou_Latn	Boudéi	L	crk_Latn	Plains Cree	M
ary_Arab	Moroccan Arabic	L	bov_Latn	Tuwuli	M	crn_Latn	El Nayar Cora	M
arz_Arab	Egyptian Arabic	L	box_Latn	Buamu	M	crq_Latn	Iyo'wujwa Chorote	H
asa_Latn	Asu (Tanzania)	M	bpr_Latn	Koronadal Blaán	M	crs_Latn	Seselwa Creole French	M
asg_Latn	Cishingini	M	bps_Latn	Sarangani Blaán	M	crz_Latn	Iyo'wujwa Chorote	M
asm_Beng	Assamese	M	boc_Latn	Boko (Benin)	M	csa_Latn	Jola-Kasa	M
ast_Latn	Asturian	L	bgg_Latn	Bago-Kusuntu	L	cso_Latn	Sochiapan Chinantec	M
ata_Latn	Pele-Ata	M	bqi_Arab	Bakhtiari	L	ctd_Latn	Tedim Chin	L
atb_Latn	Zaiwa	M	bqi_Latn	Bandial	M	cte_Latn	Tepinapa Chinantec	M
atg_Latn	Ivbie North-Okpela-Arhe	M	bqp_Latn	Busa	M	ctg_Peng	Chittagonian	M
ati_Latn	Attié	M	bra_Deva	Braj	L	cti_Latn	Tlacoatzintepec Chinantec	L
atq_Latn	Aralle-Tabulahan	H	bre_Latn	Breton	L	cto_Latn	Emberá-Catío	L
ava_Cyrl	Avanic	M	brh_Arab	Brahui	M	ctul_Latn	Chol	L
avt_Latn	Avatime	M	bri_Latn	Mokpwe	L	cuc_Latn	Usila Chinantec	M
avu_Latn	Avokaya	M	bru_Latn	Eastern Bru	M	cui_Latn	Cuiba	H
awa_Deva	Awadhi	M	brx_Deva	Bodo (India)	L	cuk_Latn	San Blas Kuna	M
awb_Latn	Awa (Papua New Guinea)	M	bac_Latn	Bassari	M	cui_Latn	Culina	H
awo_Latn	Awak	L	bah_Arab	Kati	L	cut_Latn	Teutla Cuicatec	L
ayl_Arab	Libyan Arabic	L	bej_Latn	Baangwinji	L	cux_Latn	Tepeuxila Cuicatec	L
ayo_Latn	Ayoreo	H	bek_Latn	Burushaski	L	cwa_Latn	Kabwa	H
ayp_Arab	North Mesopotamian Arabic	L	bsq_Latn	Bassa	M	cwe_Latn	Kwere	M
ayr_Latn	Central Aymara	M	bas_Latn	Akoose	M	cwt_Latn	Kuwaataay	M
ayz_Latn	Mai Brat	L	bey_Latn	Sabah Bisaya	L	cya_Latn	Nopala Chatino	M
aze_Arab	Azerbaijani	M	btd_Latn	Batak Dairi	M	cym_Latn	Welsh	H
aze_Cyrl	Azerbaijani	M	btm_Latn	Batak Mandailing	L	daa_Latn	Dangaléat	M
aze_Latn	Azerbaijani	M	bts_Latn	Batak Simalungun	M	dag_Latn	Dagbani	M
azg_Latn	San Pedro Amuzgos Amuzgo	M	btt_Latn	Bete-Bendi	M	dah_Latn	Gwahatike	M
azt_Latn	Highland Puebla Nahuatl	L	biv_Arab	Bateri	L	dan_Latn	Danish	H
bag_Latn	Tuki	L	btz_Latn	Batak Karo	M	dar_Cyrl	Dargwa	L
bak_Cyrl	Bashkir	H	bud_Latn	Ntcham	M	dav_Latn	Taita	L
bam_Latn	Bambara	M	bug_Latn	Buginese	L	dbd_Latn	Dadiya	L
ban_Latn	Balinese	M	bui_Cyrl	Bulgarian	M	dbj_Latn	Idja'an	L
bao_Latn	Waimaha	M	bun_Latn	Bulu (Cameroon)	L	dbq_Latn	Daba	H
bas_Latn	Basa (Cameroon)	L	buo_Latn	Terei	L	dec_Latn	Deccan	M
bav_Latn	Vengo	M	bux_Latn	Bokobaru	M	ddn_Latn	Dendi (Benin)	M
bax_Latn	Bamun	L	bux_Latn	Boghom	L	ded_Latn	Dedua	M
bba_Latn	Baatoum	M	bvb_Latn	Bube	L	deg_Latn	Degema	L
bbb_Latn	Barai	H	bvc_Latn	Baelelea	M	des_Latn	Desano	M
bbc_Latn	Batak Toba	M	bvz_Latn	Bauzi	H	deu_Latn	German	M
bbj_Latn	Ghomála'	M	bwq_Latn	Northern Bobo Madaré	M	dga_Latn	Southern Dagaare	M
bbi_Cyrl	Bats	L	bwu_Latn	Bura-Pabir	L	dgh_Latn	Dghwede	L
bbo_Latn	Northern Bobo Madaré	M	bwu_Latn	Buli (Ghana)	M	dgi_Latn	Northern Dagara	M
bbu_Latn	Kulung (Nigeria)	L	bxf_Latn	Bilur	L	dgl_Latn	Dagba	M
bcc_Arab	Southern Balochi	M	bxx_Latn	Bukusu	L	dgo_Deva	Dagri (individual language)	M
bcc_Latn	Southern Balochi	M	bys_Latn	U'baghara	M	dgr_Latn	Dogrib	M
bce_Latn	Bamenyam	L	byr_Latn	Baruya	H	dhi_Deva	Dhimial	M
bci_Latn	Baoulé	L	bys_Latn	Burak	L	did_Latn	Didinga	M
bcl_Latn	Central Bikol	M	biv_Latn	Medumba	M	dig_Latn	Digo	M
bcs_Latn	Kohumono	L	bys_Latn	Qaqet	L	dik_Latn	Southwestern Dinka	M

Code	Name	Res	Code	Name	Res	Code	Name	Res
dip_Latn	Northeastern Dinka	M	gyz_Latn	Gajji	L	kfb_Deve	Northwestern Kolami	M
div_Thaa	Dhivehi	L	had_Latn	Hadram	M	kff_Telu	Koya	M
dje_Latn	Zarma	M	hag_Latn	Hanga	M	kfe_Deve	Kinnauri	M
djk_Latn	Eastern Maroon Creole	M	hah_Latn	Hahon	L	kfg_Deve	Korku	L
dmk_Arab	Domaaki	L	hak_Latn	Hakka Chinese	M	kfr_Gujr	Kachhi	L
dml_Arab	Dameli	L	hao_Latn	Hakö	L	kfw_Latn	Khamar Naga	M
dnl_Latn	Dan	M	hap_Latn	Hupla	L	kfx_Deve	Kullu Pahari	M
dnt_Latn	Mid Grand Valley Dani	H	hau_Latn	Haitian	H	kha_Latn	Khasi	L
dnu_Latn	Western Dani	H	hau_Latn	Hausa	H	khg_Tibt	Khams Tibetan	M
dop_Latn	Lukpa	M	haw_Latn	Hawaiian	L	khk_Cyrl	Halh Mongolian	M
dos_Latn	Dogosé	M	hay_Latn	Haya	M	khm_Khmr	Khmer	M
dru_Latn	Rukai	L	hbb_Latn	Huiba	L	khq_Latn	Koyra Chiini Songhay	M
dsb_Latn	Lower Sorbian	L	hch_Latn	Huichol	L	khw_Arab	Khowar	M
dsh_Latn	Daasanach	H	heb_Hebr	Hebrew	H	kia_Latn	Kim	M
dtp_Latn	Kadazan Dusun	H	heh_Latn	Hehe	M	kij_Latn	Kilivila	M
dst_Latn	Toro So Dogon	M	her_Latn	Herero	L	kik_Latn	Kikuyu	M
dtv_Deve	Dotyali	L	hia_Latn	Lamang	L	kin_Latn	Kinyarwanda	H
dua_Latn	Duala	L	hif_Latn	Fiji Hindi	M	kir_Cyrl	Kirghiz	M
dug_Latn	Duruma	M	hig_Latn	Kamwe	M	kiz_Latn	Khiamiungang Naga	L
dwr_Latn	Dawro	M	hil_Latn	Hiligaynon	M	kjb_Latn	Q'anjob'al	M
dyl_Latn	Djimini Senoufo	M	hin_Deve	Hindi	H	kjc_Latn	Coastal Konjo	L
dyo_Latn	Jola-Ponyi	M	hkk_Latn	Hunjar-Kaina Ke	L	kjs_Latn	Kisar	M
dyu_Latn	Dyula	H	hla_Latn	Halia	L	kjg_Latn	Khmu	L
dzg_Latn	Dazaga	L	hib_Deve	Halbi	M	kjh_Cyrl	Khakas	M
dzo_Tibt	Dzongkha	M	hit_Latn	Matu Chin	M	kjk_Latn	Highland Konjo	M
ebu_Latn	Embu	L	hne_Deve	Chhattisgarhi	M	kkl_Latn	Kagulu	M
ego_Latn	Eggon	L	hnn_Latn	Hannuo	M	kki_Latn	Kako	M
eip_Latn	Eipomek	H	hno_Arab	Northern Hindko	M	kis_Deve	Kulung (Nepal)	M
evr_Latn	Aekopon	L	hob_Latn	Cambodian Hindustani	M	kin_Latn	Kalenjin	M
eka_Latn	Ekajuk	M	hoc_Orya	Ho	L	kis_Latn	Kalasha	L
ekk_Latn	Standard Estonian	H	hrv_Latn	Croatian	H	klu_Latn	Klao	M
eko_Latn	Koti	L	hnb_Latn	Upper Sorbian	M	kiv_Latn	Masakelynes	M
elr_Latn	Yace	M	hio_Latn	Minica Huitoto	M	kiw_Latn	Tado	L
ell_Grek	Modern Greek	H	hub_Latn	Huambisa	M	kma_Latn	Konni	M
ell_Grek_cypr1249	Cypriot Greek	L	hue_Latn	San Francisco Del Mar Huave	L	kmd_Latn	Majukayang Kalinga	M
elm_Latn	Elm	L	hul_Latn	Huli	L	kml_Latn	Tausugan Kalinga	M
emp_Latn	Northern Emberá	M	hun_Latn	Hungarian	L	kmr_Arab	Northern Kurdish	M
enb_Latn	Markweeta	M	hus_Latn	Huastec	H	kmr_Cyrl	Northern Kurdish	M
eng_Latn	English	H	hux_Latn	Huitoto	M	kmr_Latn	Northern Kurdish	M
enx_Latn	Esperanto	M	huv_Latn	San Mateo Del Mar Huave	L	kmu_Latn	Kanite	H
epo_Latn	Ese Eija	M	huz_Latn	Nipode Huitoto	L	kmy_Latn	Koma	L
ese_Latn	Central Siberian Yupik	M	hvb_Latn	Sababu	L	kna_Latn	Dera (Nigeria)	L
esu_Latn	Central Yupik	L	hwa_Latn	Hawai'i Creole English	M	knb_Latn	Lubuagan Kalinga	M
eto_Latn	Yekoi (Cameroon)	L	hwo_Latn	Hwana	L	knc_Latn	Central Kanuri	L
ets_Latn	Etne	M	hye_Armen	Armenian	L	kne_Latn	Kankanaey	M
etu_Latn	Ejagham	H	hyw_Armen	Western Armenian	M	knl_Latn	Mankanya	M
eus_Latn	Basque	H	iba_Latn	Iban	M	knj_Latn	Western Kanjober	M
evn_Cyrl	Evenki	L	ibb_Latn	Ibibio	L	knk_Latn	Kurango	M
ewe_Latn	Ewe	H	ibo_Latn	Igbo	L	knn_Deve	Konkani (individual language)	L
ewo_Latn	Ewondo	M	icr_Latn	Islander Creole English	M	kno_Latn	Kono (Sierra Leone)	M
eyo_Latn	Keiyof	L	ida_Latn	Idakho-Isukha-Tiriki	L	kog_Latn	Cogui	H
eza_Latn	Eza	M	ide_Latn	Ede Idaca	L	kol_Latn	Kol (Papua New Guinea)	L
fai_Latn	South Fali	M	idu_Latn	Idoma	L	kot_Latn	Konzo	M
fan_Latn	Fang (Equatorial Guinea)	M	ifa_Latn	Amganad Ifugao	M	kor_Hang	Korean	H
fao_Latn	Faroese	H	ifb_Latn	Batad Ifugao	M	kpo_Latn	Ikposo	M
far_Latn	Fataleka	L	ifk_Latn	Tuwali Ifugao	M	kpr_Latn	Korupun-Sela	M
fas_Arab	Persian	H	ifu_Latn	Mayoyao Ifugao	M	kps_Cyrl	Komi-Zyrian	L
fat_Latn	Fanti	L	ifv_Latn	Keley-I Kallahan	M	kpy_Cyrl	Koryak	M
faa_Latn	Nobin	M	igk_Latn	Igala	L	kps_Latn	Kupsabiny	M
fij_Latn	Fijian	M	ign_Latn	Ignaciano	M	kqa_Latn	Kaqabang	H
fil_Latn	Philippine	M	ijg_Latn	Izon	M	kqo_Latn	Eastern Krahn	M
fin_Latn	Finnish	H	ijp_Latn	Kalabari	L	kqr_Latn	Kimré	L
fip_Latn	Fipa	L	ikl_Latn	Ika	M	kqz_Latn	Kirgiz	M
flk_Latn	Kirya-Konzol	L	ikw_Latn	Ikwere	L	kqy_Ethi	Koorete	M
fir_Latn	Fulirru	H	ilb_Latn	Ila	L	krc_Cyrl	Karachay-Balkar	M
fmp_Latn	Fe'fe'	L	ilo_Latn	Iloko	M	kri_Latn	Kirati	M
fmu_Deve	Far Western Muria	L	imo_Latn	Imbongu	M	kri_Latn	Kinaray-A	M
fon_Latn	Fon	M	ina_Latn	Indonesian	L	krl_Latn	Karelian	M
fra_Latn	French	H	ina_Latn	Interlingua	L	krr_Khmr	Krung	M
frd_Latn	Fordata	H	ind_Latn	Indo	M	krs_Latn	Korbo (Sudan)	M
frj_Latn	Western Frisian	L	iou_Latn	Tuma-Irumu	M	kru_Deve	Kurukh	L
fub_Latn	Adamawa Fulfulde	M	ipk_Latn	Inupiaq	M	kru_Deve	Kurukh	L
fuc_Latn	Fulaa	L	iqw_Latn	Ikwu	M	krx_Latn	Karon	L
fue_Latn	Borgu Fulfulde	L	irf_Latn	Rigwe	M	kry_Latn	Karukh	M
ful_Latn	Fulah	H	irk_Latn	Irako	M	kub_Latn	Kuba	M
fuq_Latn	Central-Eastern Niger Fulfulde	L	ish_Latn	Isan	M	kud_Latn	Kuanua	L
fur_Latn	Nigerian Fulfulde	M	isl_Latn	Icelandic	M	kuf_Latn	Kuf	M
gag_Cyrl	Gagauz	M	iso_Latn	Isoko	L	kum_Cyrl	Kumuk	M
gag_Latn	Gagauz	M	ita_Latn	Italian	L	kuz_Arab	Kuz	M
gai_Latn	Borei	H	itv_Latn	Itawit	M	kva_Latn	Kuvaa	M
gam_Latn	Kandawo	M	itw_Latn	Ito	L	kuy_Latn	Kuy	M
gau_Telu	Mudhili Gadaba	M	itz_Latn	Itz	M	kuz_Arab	Kuz	M
gbl_Latn	Galela	M	izh_Latn	Izhma	L	kuz_Arab	Kuz	M
gbk_Deve	Gaddi	M	izr_Latn	Izere	M	kuz_Arab	Kuz	M
gbm_Deve	Garhwali	M	izz_Latn	Izii	M	kuz_Arab	Kuz	M
gbo_Latn	Northern Grebo	M	jac_Deve	Jak	M	kuz_Arab	Kuz	M
gbr_Latn	Gbagyi	L	jag_Latn	Jahaj	M	kuz_Arab	Kuz	M
gby_Latn	Gbari	L	jai_Latn	Jahaj	M	kuz_Arab	Kuz	M
gcc_Latn	Mali	L	jam_Latn	Jamaican Creole English	M	kuz_Arab	Kuz	M
gde_Latn	Gude	M	jav_Latn	Javanese	H	kuz_Arab	Kuz	M
gdt_Latn	Gudut-Gava	L	jaw_Latn	Jambi Malay	M	kuz_Arab	Kuz	M
geb_Latn	Kire	L	jbu_Latn	Jukun Takum	M	kuz_Arab	Kuz	M
gej_Latn	Gen	M	jen_Latn	Dza	L	kuz_Arab	Kuz	M
gew_Latn	Geser-Gorom	L	jic_Latn	Jica	M	kuz_Arab	Kuz	M
ggg_Arab	Gurgula	L	jiv_Latn	Jiv	M	kuz_Arab	Kuz	M
gid_Latn	Gidar	L	jmc_Latn	Machame	M	kuz_Arab	Kuz	M
gig_Arab	Goria	M	jmd_Latn	Yamdena	M	kuz_Arab	Kuz	M
gil_Latn	Gilbertese	M	jmx_Latn	Western Juxtlahuaca Mixtec	H	kuz_Arab	Kuz	M
giz_Latn	South Giziga	L	jpn_Latn	Japanese	M	kuz_Arab	Kuz	M
gjk_Arab	Kachi Koli	M	jqr_Latn	Jaquari	M	kuz_Arab	Kuz	M
gjn_Latn	Gonja	M	juk_Latn	Jukun	M	kuz_Arab	Kuz	M
gju_Arab	Gujari	L	jua_Latn	Juba	M	kuz_Arab	Kuz	M
gkn_Latn	Gokana	M	jvn_Latn	Javanese	M	kuz_Arab	Kuz	M
gid_Cyrl	Nanai	L	kaa_Cyrl	Kara-Kalpak	M	kuz_Arab	Kuz	M
gie_Latn	Irish	H	kab_Latn	Kabyle	H	kuz_Arab	Kuz	M
gig_Latn	Galician	H	kac_Latn	Kachin	M	kuz_Arab	Kuz	M
gik_Arab	Gilaki	M	kai_Latn	Karekare	L	kuz_Arab	Kuz	M
giv_Latn	Manx	L	kaj_Latn	Jju	L	kuz_Arab	Kuz	M
glw_Latn	Glavda	L	kak_Latn	Kalanguya	M	kuz_Arab	Kuz	M
gmw_Latn	Gamo	M	kam_Latn	Kamba (Kenya)	M	kuz_Arab	Kuz	M
gna_Latn	Kaana	M	kao_Latn	Xaasongaxango	M	kuz_Arab	Kuz	M
gnd_Latn	Zulgo-Gemzek	M	kaq_Latn	Capanahua	H	kuz_Arab	Kuz	M
gng_Latn	Ngangam	M	kas_Arab	Kashmiri	M	kuz_Arab	Kuz	M
gof_Latn	Gofa	M	kat_Geor	Georgian	M	kuz_Arab	Kuz	M
gog_Latn	Gogo	M	kay_Latn	Kamayura	L	kuz_Arab	Kuz	M
gol_Latn	Gola	L	kaz_Cyrl	Kazakh	M	kuz_Arab	Kuz	M
gom_Deve	Goan Konkani	L	kbd_Cyrl	Kabardian	H	kuz_Arab	Kuz	M
gor_Latn	Gorontalo	M	kbl_Latn	Kanembu	L	kuz_Arab	Kuz	M
gqr_Latn	Gor	M	kbo_Latn	Keliko	H	kuz_Arab	Kuz	M
grc_Grek	Ancient Greek (to 1453)	M	kbp_Latn	Kabiyè	M	kuz_Arab	Kuz	M
grl_Latn	Ghari	H	kqb_Latn	Kamano	M	kuz_Arab	Kuz	M
grn_Latn	Guarani	M	kbr_Latn	Kafa	M	kuz_Arab	Kuz	M
grt_Beng	Garö	M	kbt_Latn	Abadi	M	kuz_Arab	Kuz	M
gso_Latn	Southwest Gbaya	M	kby_Latn	Mangkani	M	kuz_Arab	Kuz	M
gub_Latn	Guajajára	H	kca_Cyrl	Khanty	L	kuz_Arab	Kuz	M
guc_Latn	Wayuu	M	kcg_Latn	Tyap	M	kuz_Arab	Kuz	M
gud_Latn	Yocobonté Dida	M	kcn_Latn	Kanuri	M	kuz_Arab	Kuz	M
gug_Latn	Paraguayan Guaraní	M	kco_Latn	Komo	M	kuz_Arab	Kuz	M
guh_Latn	Guhibo	H	kdc_Latn	Kutu	M	kuz_Arab	Kuz	M
gui_Latn	Eastern Bolivian Guaraní	L	kde_Latn	Makonde	M	kuz_Arab	Kuz	M
guj_Gujr	Gujarati	M	kdh_Latn	Kumam	M	kuz_Arab	Kuz	M
guk_Ethi	Gumuz	M	kdi_Latn	Karamojong	M	kuz_Arab	Kuz	M
gum_Latn	Guambiano	M	kdl_Latn	Kunda	M	kuz_Arab	Kuz	M
guo_Latn	Guayabero	M	kdt_Khmr	Kuy	M	kuz_Arab	Kuz	M
guq_Latn	Aché	M	kew_Latn	Kewer	M	kuz_Arab	Kuz	M
gur_Latn	Farefare	M	kex_Latn	Kenyan	M	kuz_Arab	Kuz	M
gus_Latn	Yanomamö	M	kfu_Latn	Kufu	M	kuz_Arab	Kuz	M
gut_Latn	Gourmanchéma	M	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
guz_Latn	Gusii	L	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gvn_Latn	Guanano	M	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gvl_Latn	Gulay	M	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gwc_Arab	Gwari	L	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gwe_Latn	Gweno	L	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gwi_Latn	Gwich'in	L	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gwr_Latn	Gware	H	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gwt_Arab	Gawar-Bati	L	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gym_Latn	Ngäbere	M	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M
gyr_Latn	Guarayu	M	kfw_Latn	Kufu	M	kuz_Arab	Kuz	M

Code	Name	Res	Code	Name	Res	Code	Name	Res
llg_Latn	Lole	L	mqy_Latn	Manggarai	L	ory_Orya	Odia	H
lle_Latn	Lele (Chad)	H	mri_Latn	Maori	L	oss_Cyrl	Ossesian	M
lne_Latn	Léle	M	mry_Cyrl	Western Mari	M	otz_Latn	Oezquital Otomi	M
lnd_Latn	Lundayeh	M	mrr_Deva	Maria (India)	L	otq_Latn	Querétaro Otomi	M
lss_Latn	Lamiso'	M	mrt_Latn	Marghi Central	L	omz_Latn	Koonzime	M
lhu_Latn	Longuda	L	mrv_Latn	Mavara	L	pab_Latn	Parecis	M
loa_Latn	Loloda	L	msh_Latn	Masikoro Malagasy	L	pad_Latn	Paumarí	M
lob_Latn	Lobi	M	msl_Latn	Sabah Malay	M	pag_Latn	Pangasinan	M
lok_Latn	Loiko	M	maw_Latn	Mansoanka	L	pan_Latn	Panpaaga	M
lom_Latn	Loma (Liberia)	M	may_Latn	Aruamu	M	pan_Guru	Panjabi	H
lon_Latn	Malawi Lomwe	M	mid_Latn	Mialang	L	pao_Latn	Northern Paiute	M
log_Latn	Lobala	M	mti_Latn	Moskona	H	pap_Latn	Papiamento	M
lrk_Arab	Loarkí	L	mto_Latn	Totontepec Mixte	H	pau_Latn	Palauan	M
ll_Latn	Lashi	M	mtr_Deva	Mewari	L	pbb_Latn	Páez	M
lsm_Latn	Saamia	M	mtu_Latn	Tututepec Mixtec	L	pbc_Latn	Patamona	M
las_Arab	Lasi	L	mtx_Latn	Tidaá Mixtec	L	pbi_Latn	Parkwa	M
lrg_Latn	Latigalian	M	mus_Latn	Mundani	L	pbs_Latn	Central Pame	L
lth_Latn	Thur	M	mug_Latn	Musgu	L	pbt_Arab	Southern Pashto	L
lto_Latn	Tsotso	L	muh_Latn	Mündü	M	pbu_Arab	Northern Pashto	M
luz_Latn	Luxembourgish	L	mul_Latn	Musi	L	pcn_Thai	Ruching Palauug	M
lua_Latn	Luba-Lulua	L	mup_Deva	Malvi	M	pcm_Latn	Nigerian Pidgin	M
luc_Latn	Atinga	M	mur_Latn	Murle	M	pet_Latn	Petats	L
lug_Latn	Ganda	M	muv_Mymr	Muthuvan	M	pez_Latn	Eastern Penan	M
luo_Latn	Luo (Kenya and Tanzania)	H	myu_Latn	Muyang	M	phl_Arab	Phalura	M
lua_Latn	Luhai	L	mvp_Latn	Duri	M	phr_Arab	Pahari-Potwari	M
lwg_Latn	Wanga	L	mvy_Arab	Indus Kohistani	M	pih_Latn	Yine	M
lwo_Latn	Luwu	M	mwd_Latn	Mün Chin	M	pij_Latn	Yom	M
lww_Latn	Lewo	M	mww_Latn	Mentawai	L	pip_Latn	Pero	M
lzz_Latn	Laz	L	mxb_Latn	Tezoatlán Mixtec	H	pir_Latn	Piratapuyo	L
maa_Latn	San Jerónimo Tecóati Mazatec	M	mxq_Latn	Juquila Mixe	M	pij_Latn	Pijin	M
mab_Latn	Yitanduchi Mixtec	M	mza_Latn	Huitepec Mixtec	L	piy_Latn	Piya-Kwocri	M
mad_Latn	Madurese	M	mxt_Latn	Jamiltepec Mixtec	M	pjt_Latn	Pitjantjatjara	H
maf_Latn	Mafa	L	mza_Latn	Mada (Cameroon)	L	pkb_Latn	Pokomo	M
mag_Deva	Magahi	M	mxv_Latn	Metlatónoc Mixtec	M	pko_Latn	Pokomo	M
mah_Latn	Marshallese	M	mxy_Latn	Southeastern Nochixtlán Mixtec	M	plk_Arab	Kohistani Shina	M
mai_Deva	Maithili	M	mya_Mymr	Burmese	H	plm_Latn	San Juan Tlacoyalco Popoloca	M
maj_Latn	Majaga De Díaz Mazatec	M	myb_Latn	Mbay	M	plt_Latn	Plateau Malagasy	M
mak_Latn	Makasar	M	myk_Latn	Mamara Senoufo	M	plw_Latn	Brooke's Point Palawano	M
mal_Mymr	Malayalam	H	myr_Cyrl	Erzya	M	pmn_Latn	Northern Pame	L
mam_Latn	Mam	M	myx_Latn	Masaaba	M	pms_Latn	Piemontese	L
maq_Deva	Chiquihuitlán Mazatec	M	myz_Latn	Macushi	M	pny_Latn	Capuan Malay	M
mau_Latn	Huaatlá Mazatec	L	mza_Latn	Santa María Zacatepec Mixtec	M	pnb_Arab	Western Panjabi	L
maw_Latn	Mampruli	M	mzi_Latn	Ixcatlán Mazatec	M	pne_Latn	Western Penan	L
max_Latn	North Moluccan Malay	M	mzj_Latn	Manyá	M	pyi_Latn	Pyin	M
maz_Latn	Central Mazahua	M	mzk_Latn	Nigeria Mambila	M	poq_Latn	Poqomam	L
mbb_Latn	Western Bukidnon Manobo	M	mzl_Latn	Mazatlán Mixe	M	pos_Latn	San Juan Atzingo Popoloca	L
mbe_Latn	Machashi	M	mzm_Latn	Mumuye	M	pot_Latn	Potomac	M
mbl_Latn	Mangseng	M	mzw_Latn	Deg	M	poqomchi	Poqomchi	H
mby_Latn	Matisigalug Manobo	M	nab_Latn	Southern Nambikuára	M	pol_Latn	Highland Popoloca	M
mca_Latn	Mbula-Bwazza	M	nag_Latn	Naga Pidgin	M	por_Latn	Portuguese	L
mcb_Latn	Machiguenga	M	nal_Latn	Nalik	M	pow_Latn	Upper Guinea Crioulo	H
mcj_Latn	Sharananahua	H	nan_Latn	Mín Nan Chinese	M	ppw_Latn	San Felipe Orlantepec Popoloca	L
mcg_Latn	Coatlán Mixe	M	nap_Latn	Neapolitan	L	ppy_Latn	Pogolo	M
mco_Latn	Makaa	M	nas_Latn	Nasioi	M	ppk_Latn	Una	H
mcq_Latn	Ede	M	naw_Latn	Nawuri	M	prf_Latn	Paranan	M
mca_Latn	Cameroon Mambila	M	nba_Latn	Ngamo	M	prk_Latn	Parak	M
mcc_Latn	Matsigen	M	ncc_Latn	Notsi	L	prq_Latn	Ashéninka Perené	L
mcd_Latn	Mda (Nigeria)	M	nch_Latn	Central Huasteca Nahuatl	M	prt_Thai	Phai	H
mda_Latn	Mbum	L	ncj_Latn	Northern Puebla Nahuatl	M	psa_Latn	Central Malay	M
mdb_Latn	Santa Lucia Monteverde Mixtec	M	ncd_Latn	Michoacán Nahuatl	M	psb_Latn	Kaulong	L
mdc_Latn	Male (Ethiopia)	M	nco_Latn	Sibe	M	psd_Latn	Central Pashto	H
mdf_Latn	Melpa	M	ncx_Latn	Central Puebla Nahuatl	L	pti_Latn	Bamban	M
mdg_Latn	Mengem	M	ndi_Latn	Samba Leko	L	pua_Latn	Western Highland Purepecha	L
mdh_Latn	Southwestern Tlaxiaco Mixtec	L	ndj_Latn	Ndamba	M	puj_Latn	Puinave	M
mdj_Latn	Meyah	M	ndo_Latn	Ndonga	M	pus_Arab	Spaspaiva	M
mek_Latn	Mekeo	M	ndp_Latn	Ndut	M	pwg_Latn	Gaspaiswa	H
mel_Latn	Central Melanau	M	ndr_Latn	Ndut	M	pwf_Latn	Pawai	M
men_Latn	Mende (Sierra Leone)	M	nds_Latn	Lutes	M	pwu_Thai	Pwo Northern Karen	M
meq_Latn	Merey	M	ndt_Latn	Ndoto	M	pwm_Latn	Quetzaltepec Mixe	M
mer_Latn	Meru	M	ndu_Latn	Nduo	M	pwn_Latn	K'iche'	H
met_Latn	Mato	M	ndv_Latn	Ndoga	M	pxm_Latn	Lambayeque Quechua	M
meu_Latn	Motu	L	ndw_Latn	Toura (Côte d'Ivoire)	M	qab_Latn	Chimborazo Highland Quichua	M
mev_Latn	Mano	M	neb_Latn	Nepali (macrolanguage)	M	qac_Latn	South Bolivian Quechua	H
mfe_Latn	Moréan	M	nep_Deva	Nepali (macrolanguage)	M	qad_Latn	North Bolivian Quechua	L
mfi_Latn	Matal	M	nfa_Latn	Dhao	L	qae_Latn	Sipacapanec	M
mfn_Latn	Wandala	M	nfr_Latn	Nafanra	M	qag_Latn	Southern Pastaza Quechua	L
mfs_Latn	North Mofu	M	nga_Latn	Ngakaa	M	qah_Latn	Yanahuasca Pasco Quechua	L
mfm_Latn	Marghi South	L	ngi_Latn	Ngizim	L	qai_Latn	Santiago del Estero Quichua	L
mfn_Latn	Cross River Mbembe	M	ngl_Latn	Lonwe	M	qaj_Latn	Sacapulteco	M
mfo_Latn	Mofu	M	ngp_Latn	Ngulu	L	qak_Latn	Tena Lowland Quichua	L
mfg_Latn	Moba	M	ngu_Latn	Guerrero Nahuatl	L	qal_Latn	Yauyos Quechua	M
mfv_Latn	Mandjak	M	nhe_Latn	Eastern Huasteca Nahuatl	M	qam_Latn	Quisco Quechua	M
mfw_Latn	Mayo	M	nbg_Latn	Tetelcingo Nahuatl	L	qan_Latn	Ambo-Pasco Quechua	L
mgd_Latn	Mababan	M	nbi_Latn	Zacatlán-Ahuacatlán-Tepetzintla Nahuatl	M	qao_Latn	Cajamarca Quechua	M
mgf_Latn	Moru	M	nbn_Latn	Central Nahuatl	M	qap_Latn	Eastern Apurímac Quechua	M
mge_Latn	Mango	M	nbj_Latn	Huacaleán Nahuatl	L	qas_Latn	Huamán-Dos de Mayo Huánuco Quechua	M
mgg_Latn	Pmpumpong	M	nbo_Latn	Noone	M	qat_Latn	Imbabura Highland Quichua	M
mgi_Latn	Makhuwa-Metto	M	nbs_Latn	Western Huasteca Nahuatl	H	qau_Latn	Cajataambo North Lima Quechua	L
mgl_Latn	Lijili	L	nbt_Latn	Isthmus-Mecayapan Nahuatl	M	qav_Latn	Margos-Yarowilca-Lauricocha Quechua	M
mgo_Latn	Meta'	M	nby_Latn	Northern Oaxaca Nahuatl	M	qaw_Latn	North Junín Quechua	M
mhi_Latn	Mex'di	M	nbz_Latn	Nias	M	qax_Latn	Napo Lowland Quechua	M
mhl_Latn	Mungaka	M	nca_Latn	Ngala	M	qay_Latn	San Martín Quechua	M
mhm_Cyrl	Eastern Mari	M	ncd_Latn	Nizao	L	qaz_Latn	Huaylla Wanca Quechua	M
mhu_Latn	Digaro-Mishmi	M	nce_Latn	Ngonbale	H	qba_Latn	Northern Pastaza Quechua	M
mhz_Latn	Maru	H	ncf_Latn	Nalca	M	qbc_Latn	Corongo Ancash Quechua	M
mib_Latn	Mat'ayan	M	ncg_Latn	Dutch	H	qbd_Latn	Huaylas Ancash Quechua	M
mib_Latn	Atatláhuca Mixtec	H	nci_Latn	Ngabla	M	qbe_Latn	Silhuas Ancash Quechua	L
mif_Latn	Ocootepec Mixtec	M	ncj_Latn	Ngic	M	qbf_Latn	Chiquián Ancash Quechua	M
mig_Latn	Mofu-Gudur	M	nck_Latn	Ninia Yali	H	qbg_Latn	Panao Huánuco Quechua	M
mig_Latn	San Miguel El Grande Mixtec	M	ncm_Latn	Orizaba Nahuatl	L	qbh_Latn	Salaasca Highland Quichua	M
mih_Latn	Chayuco Mixtec	H	ncn_Latn	Kwasio	M	qbi_Latn	Northern Conchucos Ancash Quechua	M
mii_Latn	Peñoles Mixtec	M	ncp_Latn	Nawdm	M	qbj_Latn	Southern Conchucos Ancash Quechua	M
mim_Latn	Alacatlaztala Mixtec	H	ncq_Latn	Nauende	M	qbk_Latn	Puno Quechua	L
mim_Latn	Minaangkabau	M	ncr_Latn	Ngindo	M	qbl_Latn	Cañar Highland Quichua	M
mio_Latn	Pinotepa Nacional Mixtec	M	ncs_Latn	Southern Nuni	M	qbm_Latn	Santa Ana de Tusi Pasco Quechua	L
mip_Latn	Apasco-Apoala Mixtec	M	ncd_Latn	Woun Meu	M	qbn_Latn	Arequipa-La Unión Quechua	L
miq_Latn	Miskito	M	nce_Latn	Norwegian Bokmål	M	qbo_Latn	Jauja Wanca Quechua	L
miz_Latn	Southern Puebla Mixtec	M	ncf_Latn	Nimadi	L	qbp_Latn	Logoloi	M
miz_Latn	Cacaloxtotec Mixtec	L	ncg_Latn	Nogai	M	qbq_Latn	Rabha	M
mju_Latn	Ayutla Mixtec	M	nci_Latn	Nomatsiguenga	H	qbr_Latn	Ramoašina	M
miz_Latn	Coatzacoarcos Mixtec	M	ncj_Latn	Nomatsiguenga	M	qbs_Latn	Rapanui	M
mjd_Deva	Mandēan	M	nck_Latn	Southeastern Puebla Nahuatl	M	qbt_Latn	Sampang	M
mji_Cyrl	Manan	M	ncm_Latn	Napu	M	qbu_Latn	Rawang	M
mji_Cyrl	Macedonian	H	ncn_Latn	Pedi	M	qbv_Latn	Rejang	M
mjk_Latn	Miya	M	ncp_Latn	Sierra Negra Nahuatl	M	qbw_Latn	Rendile	M
mkl_Arab	Dhatki	L	ncq_Latn	Tase Naga	M	qbx_Latn	Ringgou	L
mki_Latn	Mokole	M	ncr_Latn	Nateni	M	qby_Latn	Rohingya	M
mkn_Latn	Kupang Malay	M	ncs_Latn	Delo	M	qbz_Latn	Tarifit	M
mkg_Latn	Malagasy	H	ncd_Latn	Nyole	L	qca_Latn	Nyaturu	M
mla_Latn	Western Maninkakan	M	nce_Latn	Nupe-Nupe-Tako	M	qcb_Latn	Rajbanshi	M
mlt_Latn	Maltese	M	ncf_Latn	Nuer	M	qcc_Latn	Rangpuri	M
mmc_Latn	Michoacán Mazahua	L	ncg_Latn	Tlamancaza Nahuatl	L	qcd_Latn	Carpathanian Romani	L
mnb_Latn	North Ambrym	M	nci_Latn	Nyabwa	M	qce_Latn	Carpathanian Romani	M
mne_Latn	Muna	M	ncj_Latn	Naxi	M	qcf_Latn	Sinte Romani	M
mne_Latn	Naba	L	nck_Latn	Nyanja	M	qcg_Latn	Vlax Romani	M
mnf_Latn	Mundani	M	ncm_Latn	Nyanjole	M	qch_Latn	Vlax Romani	M
mnj_Beng	Manipuri	M	ncn_Latn	Nyoro	M	qci_Latn	Ronga	L
mkn_Latn	Mandinka	M	ncp_Latn	Nyungwe	M	qcl_Latn	Ranglong	M
mny_Mymr	Mon	M	ncq_Latn	Nzima	M	qcm_Latn	Tae'	M
mxx_Latn	Manikion	H	ncr_Latn	Obo Manobo	M	qcn_Latn	Rombo	L
moa_Latn	Mwan	M	ncs_Latn	Occitan	M	qco_Latn	Romanash (Sursilvan)	M
mog_Latn	Mongondow	M	ncd_Latn	Od	M	rod_Latn	Romblomanon	M
moj_Latn	Mopán Maya	M	nce_Latn	Oduai	L	roa_Latn	Romanian	H
mos_Latn	Mossi	M	ncf_Latn	Khana	M	rob_Latn	Rotokas	M
moz_Latn	Molima	M	ncg_Latn	Northwestern Ojibwa	M	rop_Latn	Kriol	M
moz_Latn	Mukulu	M	nci_Latn	Northwestern Ojibwa	M	ror_Latn	Waima	M
mpg_Latn	Marba	M	ncj_Latn	Oku	M	ros_Latn	Ratahan	L
lpmn_Latn	Yosondúa Mixtec	H	nck_Latn	Mochi	M	rub_Latn	Gangu	M
mpa_Latn	Migabac	M	ncm_Latn	South Tairora	H	ruc_Latn	Ruuli	L
mpb_Latn	Misima-Panaeati	M	ncn_Latn	Lingao	M	rud_Latn	Luguru	M
mpc_Latn	Mbukus	M	ncp_Latn	Tolhoon O'odham	M	rug_Latn	Roviana	M
mpd_Latn	Momuna	H	ncq_Latn	Orma	M	ruh_Latn	Rundi	M
mqi_Latn	Mamasá	M	ncr_Latn	Oromo	M	rus_Cyrl	Russian	H
mqj_Latn	Morone	M	ncs_Latn	Ormuri	M	rwn_Latn	Amba (Uganda)	L

Code	Name	Res	Code	Name	Res	Code	Name	Res
rwv_Deve	Marwari (India)	L	tfr_Latn	Teribe	M	vmz_Latn	Mazatlán Mazatec	L
sab_Latn	Buglere	M	tgc_Latn	Tigak	L	vro_Latn	Võro	L
sag_Latn	Sango	M	tgj_Latn	Tagin	H	vun_Latn	Vunjo	M
sah_Cyrl	Yakut	M	tgk_Cyrl	Tajik	H	vut_Latn	Vute	L
saj_Latn	Sahu	M	tgj_Latn	Tagalog	L	wal_Ethi	Wolaytta	M
saq_Latn	Samburu	M	tgo_Latn	Sudest	M	wal_Latn	Wolaytta	M
sas_Latn	Sasak	M	tgp_Latn	Tangao	M	wap_Latn	Wapishana	H
sau_Latn	Saleman	M	thg_Deve	Thaai	L	war_Latn	Waray (Philippines)	M
say_Latn	Saya	L	the_Deve	Chitwania Tharu	L	waw_Latn	Waiwai	M
sba_Latn	Ngambay	M	thk_Latn	Tharaka	M	way_Latn	Wayana	M
sbd_Latn	Southern Samo	M	thl_Deve	Dangaura Tharu	M	wbo_Latn	Wob	M
sbl_Latn	Botolan Sambal	M	thq_Deve	Kochila Tharu	L	wbi_Latn	Wakhi	L
sbn_Arab	Sindhi Bhil	L	thr_Deve	Rana Tharu	L	wbr_Deve	Wagdi	L
sbp_Latn	Sangu (Tanzania)	H	thv_Ting	Tahaggart Tamahaq	L	wci_Latn	Waci Gbe	L
scl_Latn	Sakachep	M	tig_Ethi	Tigré Tanna	L	wec_Latn	Wemiale	L
scd_Deve	Sadri	M	tih_Latn	Timugon Murut	M	wes_Latn	Cameroon Pidgin	L
scl_Arab	Shina	L	tik_Latn	Tikar	M	wja_Latn	Waja	L
scn_Latn	Sicilian	L	tio_Latn	Teop	L	wji_Latn	Wajri	L
scs_Latn	Scots	L	tir_Ethi	Tigrinya	H	wlo_Latn	Wolio	L
sda_Latn	Toraja-Sa'dan	M	tkg_Latn	Tesaka Malagasy	M	wlx_Latn	Wali (Ghana)	M
sdo_Latn	Bukar-Sadung Bidayuh	L	tkr_Latn	Tsakhur	L	wmw_Latn	Mwani	M
sei_Latn	Semai	L	tkd_Deve	Kathoriya Tharu	L	wob_Latn	Wo Northern	M
seh_Latn	Sema	M	tib_Latn	Tobelo	H	wof_Latn	Gambian Wolof	L
sei_Latn	Seri	L	til_Latn	Tilingit	M	wol_Latn	Wolof	M
ses_Latn	Koyraboro Senni Songhai	M	tij_Latn	Talinga-Bwisi	M	wsg_Teiz	Adilabad Gondi	M
sew_Latn	Seceyo	H	tjp_Latn	Filomena Mata-Coahuilán Totonac	M	wwa_Latn	Waama	M
sgb_Latn	Mag-antsi Ayta	M	tiy_Latn	Talysh	M	xal_Cyrl	Kalmyk	M
sgj_Deve	Surgujia	M	tmc_Latn	Tumak	M	xdy_Latn	Malayic Dayak	L
sgw_Ethi	Sebat Bet Gurage	M	tmf_Latn	Toba-Maskoy	H	xed_Latn	Xerénte	L
shi_Latn	Tachelhit	M	tna_Latn	Tacana	M	xhe_Arab	Khetrani	L
shk_Latn	Shilluk	M	tng_Latn	Tobanga	M	xho_Latn	Xhosa	M
shn_Mymr	Shan	M	tnk_Latn	Kwamera	M	xka_Arab	Kalkoti	L
sho_Latn	Shanga	M	tnb_Latn	North Tanna	M	xkl_Latn	Mainstream Kenyah	L
shp_Latn	Shipibo-Conibo	M	tnp_Latn	Whitesands	L	xmf_Geor	Mingrelian	L
sid_Latn	Sidamo	M	tnr_Latn	Ménik	H	xmnd_Latn	Madang Malay	M
sig_Latn	Paasaal	M	tns_Latn	Tontemboan	H	xmv_Latn	Antankarana Malagasy	M
sil_Latn	Tumuhlung Sisaala	M	tob_Latn	Toba	M	xnj_Latn	Ngoni (Tanzania)	M
sin_Sinh	Sinhala	L	toc_Latn	Coyutla Totonac	L	xnr_Deve	Kangri	M
sip_Tibt	Sikkimese	L	toh_Latn	Gitonga	M	xog_Latn	Xog	M
siw_Latn	Siwal	L	tok_Latn	Tok Pisin	L	xon_Latn	Konkomba	M
sja_Latn	Epena	M	tom_Latn	Tombulu	M	xpe_Latn	Liberia Kpelle	L
sjm_Latn	Mapun	M	top_Latn	Papantla Totonac	L	xrb_Latn	Eastern Karaboro	M
sjp_Deve	Surjapuri	L	tos_Latn	Highland Totonac	M	scb_Latn	Sambal	M
skf_Latn	Sik'ak	L	tpi_Latn	Tok	H	xsm_Latn	Kasem	M
skr_Arab	Sakalava Malagasy	L	tpl_Latn	Tlacoapa Me'phaa	L	xsr_Deve	Sherpa	M
skr_Arab	Saraiki	L	tpm_Latn	Tampulma	M	xsu_Latn	Xanuna	M
sid_Latn	Sissala	M	tpq_Latn	Pisaflores Tepehua	L	xta_Latn	Alcozauca Mixtec	L
sik_Latn	Slovak	M	tpu_Latn	Tlachichilco Tepehua	M	xtd_Latn	Diuxi-Tilantongo Mixtec	H
slu_Latn	Selaru	L	tiput_Latn	Tinpuiz	L	xtl_Latn	Sinicahua Mixtec	M
slv_Latn	Slovenian	H	tqp_Latn	Tomoiop	L	xtm_Latn	Magdalena Peñaasco Mixtec	M
smi_Latn	Central Sama	M	trc_Latn	Copala Triqui	M	xtn_Latn	Northern Tlaxiaco Mixtec	H
smo_Latn	Samoan	M	tri_Latn	Trió	L	xty_Latn	Cuyamecalco Mixtec	M
sna_Latn	Shona	M	trn_Latn	Trinitario	M	xua_Tam1	Alu Kurumba	L
snb_Latn	Sinauogoro	L	trp_Latn	Kok Borok	M	xuo_Latn	Kuo	M
snd_Arab	Sindhi	M	trq_Latn	San Martín Itunyoso Triqui	M	yaa_Latn	Yaminahua	M
sne_Latn	Bau Bidayuh	M	trs_Latn	Chichahuaxtla Triqui	L	yad_Latn	Yadua	M
snk_Latn	Soninke	L	trv_Latn	Sediq	M	yaj_Latn	Yalunka	M
snm_Latn	Siona	H	trw_Arab	Torwali	M	yam_Latn	Yamba	M
snp_Latn	Siane	M	tsn_Latn	Tswana	L	yao_Latn	Yao	L
snv_Latn	Sa'ban	M	tso_Latn	Tsonga	M	yag_Latn	Yagui	M
snw_Latn	Selee	M	tsz_Latn	Purepecha	M	yas_Latn	Nugunu (Cameroon)	M
soi_Latn	Solano	L	tte_Latn	Tekkitekko	H	yat_Latn	Yambeta	M
som_Latn	Somali	H	tti_Latn	Tawallammat Tamaqaj	M	yaw_Latn	Yavgbet	M
soy_Latn	Miyobe	M	ttj_Latn	Tooro	M	yay_Latn	Agwagwune	L
spa_Latn	Spanish	H	ttg_Latn	Tawallammat Tamaqaj	M	yaz_Latn	Lokaa	H
spg_Latn	Supyire Senoufo	L	ttu_Latn	Tera	L	yba_Latn	Yala	M
sps_Latn	Saposa	L	ttu_Latn	Torau	L	ybb_Latn	Yemba	M
spy_Latn	Sabaot	M	tue_Latn	Tuyuca	M	ycl_Latn	Lolopo	H
srd_Latn	Logudorese Sardinian	M	tuf_Latn	Central Tunebo	H	ycn_Latn	Yucuna	M
srd_Latn	Sardinian	L	tul_Latn	Tupuri	L	ydd_Hebr	Eastern Yiddish	M
sri_Latn	Siriano	M	tuk_Arab	Turkmen	M	ydg_Arab	Yidgha	L
srn_Latn	Saramaccan	M	tuk_Latn	Turkmen	M	yee_Miym	Ravula	M
srn_Latn	Sranan Tongo	M	tul_Latn	Turkmen	L	yek_Latn	Tarok	M
sro_Latn	Campidanese Sardinian	L	tuo_Latn	Tucano	M	yem_Latn	Kyankpa	M
srp_Cyrl	Serbian	H	tuq_Latn	Tedaga	L	yfa_Latn	Yakan	M
srp_Latn	Serbian	L	tur_Latn	Turkish	M	yji_Latn	Angguruk Yali	M
srz_Deve	Sirmauri	M	tur_Latn	Turkana	L	yor_Latn	Yoruba	H
ssi_Arab	Sansi	L	tuy_Latn	Tugen	L	yre_Latn	Yaouré	M
ste_Latn	Liana-Seti	L	tvo_Latn	Tidore	L	yua_Latn	Yucateco	M
stin_Latn	Owa	H	twi_Latn	Twene	L	yue_Hans	Yue Chinese	H
stp_Latn	Southeastern Tepehuan	M	twv_Latn	Sedoa	H	yue_Hant	Yue Chinese	M
sua_Latn	Sulka	L	twb_Latn	Western Tawbuid	M	yuz_Latn	Yuracare	M
suc_Latn	Western Subanon	M	tws_Latn	Tewa (Indonesia)	L	yva_Latn	Yawa	M
suk_Latn	Sukuma	M	twu_Latn	Ternau	M	zaa_Latn	Sierra de Juárez Zapotec	M
sun_Latn	Sundanese	H	txa_Latn	Tombonuo	M	zab_Latn	Western Tlacoilua Valley Zapotec	M
sur_Latn	Mwaghavul	M	txq_Latn	Tii	M	zac_Latn	Ocotlán Zapotec	L
sus_Latn	Susu	L	txs_Latn	Tonsea	L	zad_Latn	Cajonos Zapotec	M
svu_Latn	Puroik	L	txy_Latn	Kayapó	L	zai_Latn	Zai	M
suz_Deve	Sunwar	M	tye_Latn	Tanoso Malagasy	M	zam_Latn	Isthmus Zapotec	L
sva_Geor	Svan	M	tyh_Latn	Kyanga	L	zao_Latn	Miahuatlán Zapotec	M
swa_Latn	Swedish	H	tzh_Latn	Tzeltal	M	zaq_Latn	Aloápam Zapotec	H
swh_Latn	Swahili (individual language)	H	tzi_Latn	Tz'utujil	H	zar_Latn	Rincón Zapotec	M
swv_Deve	Shekhawati	L	tzo_Latn	Tzotzil	M	zas_Latn	Santo Domingo Albarradas Zapotec	M
szb_Latn	Suba	H	ubu_Latn	Buhí'non Bikol	M	zav_Latn	Yatzachi Zapotec	L
szn_Latn	Sangir	M	ubu_Latn	Umbu-Ungu	H	zaw_Latn	Mitla Zapotec	M
sga_Latn	Siang	L	udl_Latn	Wuzlam	L	zca_Latn	Coatecas Altas Zapotec	M
sgl_Latn	Sylheti	L	udm_Cyrl	Udmurt	M	zga_Latn	Kinga	H
sga_Latn	Semelai	M	udu_Latn	Uduk	M	zim_Latn	Mesme	M
sgy_Latn	Sakizaya	M	uig_Arab	Uighur	M	ziw_Latn	Zigua	M
tac_Latn	Lowland Tarahumara	M	uig_Cyrl	Uighur	M	zmx_Latn	Mbandja	M
taj_Deve	Eastern Tamang	M	uki_Orya	Kui (India)	L	zne_Latn	Zande (individual language)	M
tam_Tam1	Tamil	H	ukr_Cyrl	Ukrainian	H	zoc_Latn	Copainalá Zoque	L
tan_Latn	Tangale	L	ukv_Latn	Kuku	L	zoh_Latn	Chimalapa Zoque	L
teo_Latn	Yami	H	umb_Latn	Umbundu	M	zor_Latn	Rayón Zoque	M
tap_Latn	Taibwa	M	upv_Latn	Uripiv-Wala-Rano-Atchin	M	zos_Latn	Francisco León Zoque	M
taq_Latn	Tamasheq	M	ura_Latn	Urarina	M	zpc_Latn	Choapan Zapotec	H
tar_Latn	Central Tarahumara	M	urb_Latn	Urubú-Kaapor	M	zpg_Latn	Guevea De Humboldt Zapotec	M
tat_Cyrl	Tatar	M	urd_Arab	Urdu	H	zpi_Latn	Santa María Quiérolani Zapotec	M
tav_Latn	Tatuyo	H	urd_Deve	Urdu	M	zpl_Latn	Lachixío Zapotec	M
tay_Latn	Atayal	L	urd_Latn	Urdu	M	zpm_Latn	Mixtepec Zapotec	M
tbc_Latn	Takia	M	urh_Latn	Urhobo	L	zpo_Latn	Amatlán Zapotec	M
tbf_Latn	Mandari	L	urk_Thai	Urak Lawoi'	M	zpt_Latn	San Vicente Coatlán Zapotec	M
tbi_Latn	Tboli	H	urk_Thai	Urak Lawoi'	M	zqa_Latn	Quioquitani-Quieri Zapotec	M
tby_Latn	Tatuyo	M	urk_Thai	Urak Lawoi'	M	ztg_Latn	Xanagüya Zapotec	L
tca_Latn	Ticuna	M	uzb_Cyrl	Uzbek	H	ztl_Latn	Loxicha Zapotec	L
tcb_Latn	Ticuna	M	uzb_Latn	Uzbek	H	zts_Latn	Quioquitani-Quieri Zapotec	M
tcc_Latn	Datoga	M	uzn_Latn	Northern Uzbek	M	ztu_Latn	Tilquiapan Zapotec	L
tcd_Latn	Malmaltepec Me'phaa	L	vag_Latn	Vagla	L	ztu_Latn	Stiña Zapotec	L
tcy_Miym	Tulu	L	vah_Deve	Varhadi-Nagpuri	M	zty_Latn	Yatee Zapotec	M
tcz_Latn	Thado Chin	L	vai_Latn	Vai	L	zul_Latn	Zulu	H
tdj_Latn	Tajio	L	var_Latn	Huarifo	L	zyb_Latn	Yongbei Zhuang	M
tdn_Latn	Tondano	L	ver_Latn	Mom Jango	M	zyz_Latn	Zyphie Chin	L
tdx_Latn	Tandroy-Mahafaly Malagasy	M	vid_Latn	Vidudua	M	zza_Latn	Zaza	L
ted_Latn	Tepo Krumen	M	vie_Latn	Vietnamese	H			
tee_Latn	Huehuetla Tepehua	M	vii_Latn	Vili	M			
tel_Telu	Telugu	H	vmc_Latn	Juxtlahuaca Mixtec	L			
tem_Latn	Timne	M	vmj_Latn	Ixtayutla Mixtec	L			
teo_Latn	Teso	M	vmn_Latn	Mitlatlan Mixtec	L			
ter_Latn	Terenó	M	vmp_Latn	Soyaltepec Mazatec	M			
tew_Latn	Tewa (USA)	M	vmw_Latn	Makhuwa	L			
tex_Latn	Tennet	M	vmy_Latn	Ayautla Mazatec	M			

Table 25 Full list of languages supported by Omnilingual ASR, including language code, English name, and resource level (Low, Medium, High).

B WER Filtering

WER-thresholds were used to filter out samples likely to be of low quality from the Omnilingual ASR Corpus ASR dataset. Values ranged from 150 to 250 WER. These were determined qualitatively and selected to filter out samples with obviously misaligned audio/text. For example:

Reference:

okoro ekwup mmotima nson wo mawanne ochike machip akpan pimoruku bebogy

Hypothesis:

okoro ekwu otok kpene kpe fu bok obo mo tim so woma wane mo chike ma achit
akpe pa mo orugo be boya bep be bae bake bonga akpe pe nok boya

Reference:

en sa w konn sa k pase

Hypothesis:

en fò w konn sa k pase n ap tou benefisye yon staj men m byen kwè so kò
kòman kote sa ye lankò menm chak ki bay bon moun yo wi me nm ja ou ka

Reference:

enh se fèt dè mè se fèt ou ankò

Hypothesis:

elepicit m konnen lepichit m konnen wi m konnen demis li rele en skisoe
bon tetout fason pann fèt aa o byen pete ye e fèe fèt b èmè pis fèt ou ankò

In the above examples, it is clear in listening to the audio that the hypotheses generated by our model are more accurate than the reference texts, so we filtered such examples out.

C Prompts and Guidelines for Commissioned Data Collection

This section contains the recording prompts and transcription guidelines for our commissioned data collection.

C.1 Recording guidelines

- Please record in a quiet environment.
- During the recording, please refrain from:
 - touching the microphone,
 - blowing into the microphone,
 - moving things around that are close to the recording device.
- Please refrain from clearing your throat, coughing, sneezing, or making any loud sounds during the recording.
- Please refrain from eating or drinking during the recording.
- Please speak in a natural, normal voice.
- Please speak at a normal pace and not too quickly or too slowly.
- If you encounter names and words that are in a different language (for example, an English name when you are speaking Swahili), please do your best to pronounce the name as you normally would in the target language.
- Please refrain from sharing any personally identifiable information in the recordings, whether it pertains to you or others. Personally identifiable information includes:
 - Full name
 - Phone number
 - Home address
 - Email address or other account identifiers, such as social media handles
 - Passport number
 - Social Security Number or analogous identification numbers

- Health information
- Sexual orientation
- Political affiliation
- Any other analogous information

C.2 Transcription guidelines

Your job is to transcribe exactly what was said in the recording, including a representation of all the disfluencies and noises it contains.

- If the recording contains grammatical mistakes, these should not be corrected in the transcription.
- The only characters allowed in the transcription are letters of the given language, punctuation and the set of special tags specified below.
- (Updated) Wherever possible and if this is applicable to your language, please use punctuation in transcripts as you would normally do in your written language. Please also capitalize the beginnings of new sentences if applicable.

Numbers and acronyms.

- Numbers should be spelled out in words. They should not be written in the numeral system.
 - **Incorrect:** *I walked exactly 2017 steps.*
 - **Correct:** *I walked exactly two thousand seventeen steps.*
- Acronyms should be written as they are normally written in the language, following standard capitalization rules. They should not be transcribed phonetically.
 - **Incorrect:** *They were arrested by the eff bee eye last Thursday.*
 - **Correct:** *They were arrested by the FBI last Thursday.*

Punctuation and symbols

- Use the punctuation that is appropriate for writing in the given language.
- Symbols for currencies, percentages, etc. should be avoided, and should instead be spelled out.
 - **Incorrect:** *This bag cost me only \$10!*
 - **Correct:** *This bag cost me only ten dollars!*

Special tags The following special tags should be used to mark disfluencies, fillers, and other types of non-verbal content.

Tag	Meaning
<laugh>	The sound of laughter.
<hesitation>	A hesitation sound, often used by speakers while thinking of the next thing to say. In English, some common hesitation sounds are “err”, “um”, “huh”, etc.
<unintelligible>	A word or sequence of words that cannot be understood.
<noise>	Any other type of noise, such as the speaker coughing or clearing their throat, a car honking, the sound of something hitting the microphone, a phone buzzing, etc.

Table 26 Special tags used for transcription

- Tags should be inserted in the transcription at the appropriate location, and should be separated from the other content by spaces; for example:
 - *And then I <noise> went on holiday.*
 - *Well, <noise> <laugh> it wasn't exactly a holiday <laugh>*
- When we speak, we often insert hesitations while thinking of the next idea we want to say. Some common hesitations in English are “err”, “um” and “uh”. Since these hesitations can vary significantly in the exact sounds and length used, and often there are no clear rules on how they should be written, for

this project they should all be represented using the tag `<hesitation>`. Only this tag should be used. You should not attempt to transcribe hesitations using letters, such as “err”.

Word segments, false starts and repeated words.

- Spontaneous speech naturally contains false starts where only a fragment of a full word is produced. For these instances, please transcribe to the best of your ability the word fragment and attach a hyphen at the end of the word (-) to indicate the word is a false start.
 - *His name is Jo- Jona- Jonathan.*
- Sometimes speakers will repeat a word or word fragment multiple times. This should be transcribed too.
 - *And then I went to the the the bed- the bedroom.*

Grammatical mistakes and colloquialisms.

- Spontaneous speech will naturally contain grammatical mistakes. These should not be corrected when transcribing. The transcription should reflect the spoken content exactly.
- Speakers may use colloquialisms (such as, in English, “gonna”, “cuz”, etc.) which may not be considered formally correct. These should be transcribed as they are, and not changed to their more formal equivalents.

D Quality Assurance (QA) Guidelines

In this appendix, we detail the guidance provided to perform quality assurance (QA).

D.1 Speech recording error taxonomy

Table 27 shows the definitions used for each of the error categories. More broadly, QA technicians were asked to pay particular attention to the following speech recording issues:

- General audio quality issues (e.g., volume is too low, speech is inaudible, there is constant background noise or heavy static, files seem systematically cut off before the end)
- Ad hoc noises (e.g., rooster crowing, mechanical noises, bells or phones ringing, very long silences or pauses)
- Other human voices (e.g., people talking in the background in the same language, or more problematic, in a different language)
- The speaker responds to the prompts in a pivot language, not in the expected language (prompts were translated into a number of high-resource pivot languages and it can happen that the speaker will respond to the prompts in the same language as the prompts instead of responding in their native language)

D.2 Transcript error taxonomy

Table 28 shows the definitions used for each of the error categories. More broadly, QA technicians were asked to pay particular attention to the following transcript issues:

- General transcription issues (e.g., the transcript does not match the audio file at all, the transcript is in an unexpected writing system, the transcript is in the International Phonetic Alphabet, the transcript is missing words, the transcript is much shorter or longer than it should be)
- Transcription issues that are specific to a language (e.g., a few non-Unicode-compliant characters have been used)
- Issues related to the use of event-marking tags (a specific tag set has been defined by the project team; Table 26)

Category	Critical example	Minor example
Human vocal noise	Second voice in the background Singing in the background	N/A (This error is always critical)
Cutoff	Speech is cut off at either end of the recording	N/A (This error is always critical)
Background noise	Rooster crowing Street noise, car honking Bird chirping Strong wind	Occasional mild coughing Occasional mild coughing Mild breathing sound
Audio Glitches	Serious glitches that break up speech	Mild glitch happens in between speech
Static noise	Strong static noise that affects intelligibility	Mild static noise that does not affect speech
Low volume	Cannot hear the speech clearly in the max volume setting	Lower than normal but still audible at max volume
Inconsistent volume	Volume changing drastically	Occasional soft voice
Muffled voice	Muffled voice sounds like talking behind a curtain	Audio is not crisp but does not affect intelligibility
Echo	Strong echo like speaking in a cave or tunnel such that it compromises the intelligibility of words	Mild echo in non-studio environment
Microphone Noise	Any hissing, plosive, popping noise that breaks the speech	Mild pop noise when turning on/off the recorder
Pause / Silence	Long pauses - If at the start or end of speech and above 2s - If at the middle of speech and above 5s - If more than $\frac{1}{3}$ of the audio is made up of leading/trailing silence or intra-sentential silence (excluding normal pauses between words)	Short pauses when speaker is thinking
Unnatural speech	Consistent stutter or mumbling Extremely not fluent, words uttered individually Whisper Feels like someone reading / monotonous speech	Occasional repeated words and syllable

Table 27 Description of all error categories used for speech recording in-depth quality assurance.

Category	Critical example	Minor example
Mismatch	Transcript file does not match the audio at all (either in content or in length)	N/A (mismatch is critical)
Wrong writing system	The transcript does not use the expected writing system The transcript is in IPA or other phonetically-based system Different writing standard, inconsistency in the spelling (the same word spelled in different ways)	N/A (writing system is critical)
Wrong tags	The transcript includes made-up tags Tags are not used adequately (e.g., <noise> instead of <hesitation>)	N/A (all mistaggings are critical)
Numbers	The presence of numbers written in digits	(N/A writing digits is critical)
Incomplete	The transcript is abridged rather than verbatim The transcript consistently misses words	The transcript seems to sometimes be missing a word or two
Inconsistent tagging	The tag set being used is compliant but the transcriber consistently switches between tags for the same audio events	A few tags show inconsistency, especially for borderline audio events

Table 28 Description of all error categories used for transcript in-depth quality assurance.