Journal of Machine Learning Research 25 (2024) 1-52 Submitted 10/23; Revised 1/24; Published 3/24

Scaling Speech Technology to 1,000+ Languages

Vineel Pratap* **Andros Tjandra*** **Bowen Shi*** **Paden Tomasello**
Arun Babu **Sayani Kundu†** **Ali Elkahky‡** **Zhaoheng Ni**
Apoorv Vyas **Maryam Fazel-Zarandi** **Alexei Baevski** **Yossi Adi**
Xiaohui Zhang **Wei-Ning Hsu** **Alexis Conneau§** **Michael Auli***

FAIR, Meta

Editor: Alexander Clark

Abstract

Expanding the language coverage of speech technology has the potential to improve access to information for many more people. However, current speech technology is restricted to about one hundred languages which is a small fraction of the over 7,000 languages spoken around the world. The Massively Multilingual Speech (MMS) project increases the number of supported languages by 10-40x, depending on the task while providing improved accuracy compared to prior work. The main ingredients are a new dataset based on readings of publicly available religious texts and effectively leveraging self-supervised learning. We built pre-trained wav2vec 2.0 models covering 1,406 languages, a single multilingual automatic speech recognition model for 1,107 languages, speech synthesis models for the same number of languages, as well as a language identification model for 4,017 languages. Experiments show that our multilingual speech recognition model more than halves the word error rate of Whisper on 54 languages of the FLEURS benchmark while being trained on a small fraction of the labeled data. The MMS models and tooling for data pre-processing are available at `https://github.com/pytorch/fairseq/tree/master/examples/mms`.

Keywords: multilingual speech processing, self-supervised learning, language expansion, neural networks

1. Introduction

Speech technology has made much progress over the past decade (Chan et al., 2015; Graves et al., 2006; Baevski et al., 2020b; Radford et al., 2022) and has been integrated into many consumer products, such as home assistants and smartphones. Despite this progress, speech technology is still absent for the vast majority of the over 7,000 languages spoken around the world (Lewis et al., 2016). Moreover, many of these languages are at risk of disappearing by the end of this century and the narrow language coverage of current technology may contribute to this trend (Bromham et al., 2021).

Speech models have traditionally been built by training models on large amounts of labeled training data which is only available for a small number of languages. More recently,

*. Core Team. Corresponding Authors: `{vineelkpratap,michaelauli}@meta.com`.

†. JPMorgan Chase. Work done while at Meta AI.

‡. Apple. Work done while at Meta AI

§. OpenAI. Work done while at Meta AI

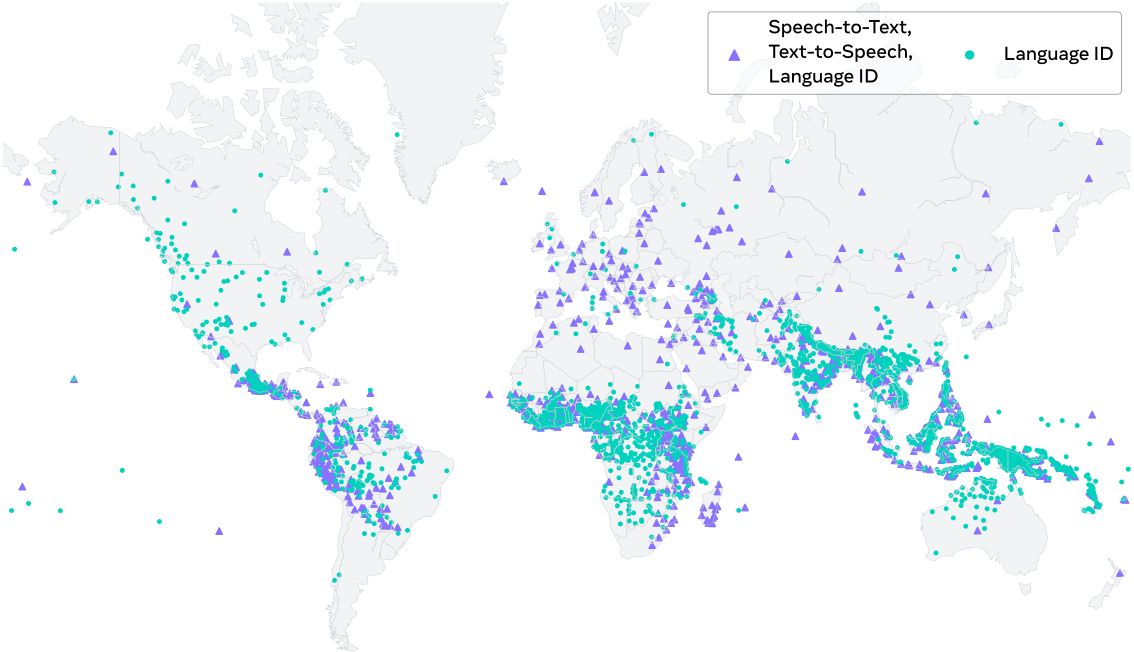


Figure 1: Illustration of where the languages supported by MMS are spoken around the world: MMS models support speech-to-text and text-to-speech for 1,107 languages as well as language identification for 4,017 languages.

self-supervised speech representations have dramatically lowered the amount of labeled data required to build speech systems (van den Oord et al., 2018; Schneider et al., 2019; Baevski et al., 2020b). But despite this progress, prominent recent work still only supports about 100 languages (Radford et al., 2022; Zhang et al., 2023a). Prior work attempting to scale beyond 100 languages suffers under a very high error rate (Black, 2019; Li et al., 2022). This means that speech technology works disproportionally well for a small number of languages and for the remaining 98%+ languages, there are models with very poor performance or there is not speech technology at all.

To help level the playing field, we build a new dataset comprising a moderate amount of labeled data for 1,107 languages and another dataset of unlabeled speech in 3,809 languages (§3). We leverage this data to pre-train wav2vec 2.0 models supporting several times more languages than any known prior work (§4) and then fine-tune these models to build a multilingual speech recognition model supporting 1,107 languages (§5), language identification models for 4,017 languages (§6) and text-to-speech models for 1,107 languages (§7).

The Massively Multilingual Speech (MMS) project aims to expand speech technology to many more people and we hope that it can be a small contribution to preserving the languages diversity of this world. Figure 1 illustrates the location where the languages supported in this work are spoken and which tasks we cover for each language.

2. Related Work

Multilingual Speech Datasets. Current multilingual speech datasets with transcriptions are drawn from a variety of domains, including Wikipedia (Ardila et al., 2020; Conneau et al., 2022), political speech (Wang et al., 2021), audiobooks (Pratap et al., 2020c) to name a few. They are limited to about 100 languages with some datasets only containing data for European languages (Wang et al., 2021; Pratap et al., 2020c). Multilingual datasets without transcriptions include VoxLingua107 (Valk and Alumäe, 2020) spanning data in 107 languages, or VoxPopuli (Wang et al., 2021) which contains large amounts of unlabeled data for European languages. Leong et al. (2022) covers 56 low-resource languages and 428 hours of data.

Compared to prior work utilizing read versions of the New Testament Black (2019), MMS covers many more languages (58% more for MMS-lab and over five times more for MMS-unlab) and our alignments are of much higher quality which leads to better models as we show in §3.3.1. We also use the resulting data to train self-supervised models, build speech recognition systems as well as language identification models while as Black (2019) focused on speech synthesis.

Multilingual Automatic Speech Recognition (ASR). Prior work on multilingual speech recognition includes both non-neural methods (Burget et al., 2010; Lin et al., 2009), hybrid neural and HMM models (Heigold et al., 2013) and more recently neural systems (Cho et al., 2018; Toshniwal et al., 2018; Kannan et al., 2019; Li et al., 2019). Li et al. (2021) built multilingual ASR for up to 15 languages, Pratap et al. (2020a); Lugosch et al. (2022); Tjandra et al. (2022b) trained and explored several strategies for 50+ languages. There is also the recent ML-SUPERB challenge which provides an evaluation framework for up to 143 languages for ASR and LID Shi et al. (2023).

Whisper (Radford et al., 2022) mines 680K hours of data from the web and their model supports the transcription of 99 different languages. Zhang et al. (2023a) trained a multilingual ASR model based on YouTube audio data and successfully scaled to 100 languages. A notable exception is Li et al. (2022) who created ASR for 1,909 languages by mapping the phoneme-like output of an eight language multilingual model to appropriate phonemes for the language of interest. In contrast, MMS uses actual paired speech and text for over 1,100 languages and present a comparison to their approach below (§3.3.2).

Spoken Language Identification (LID). Whisper (Radford et al., 2022) also supports language identification and can distinguish between 99 different languages. Fan et al. (2021) utilizes wav2vec 2.0 for language identification using the API17-OLR dataset that consists of ten Asian languages. Later, Tjandra et al. (2022a) demonstrated that a cross-lingual self-supervised model can improve language identification performance by training a language identification model for 26 languages using a proprietary dataset. Babu et al. (2022) fine-tunes a pre-trained model to perform LID for 107 languages using the VoxLingua-107 dataset (Valk and Alumäe, 2020). In this work, we scale the number of languages to over 4,000 which to our knowledge is the broadest coverage spoken language identification model so far.

Multilingual Text-To-Speech (TTS). Speech synthesis has been undergoing a transition from controlled settings to the generation of more diverse speech, with multilinguality being a crucial aspect (Casanova et al., 2022; Zhang et al., 2023b). However, the lack of multilingual training data, particularly for low-resource languages, presents a common obstacle in scaling TTS to more languages. To address data scarcity, prior work explored various approaches including byte encoding to unify text representations which was evaluated on English, Spanish, and Chinese (Li et al., 2019) Further studies explored other input representations, including phonemes (Zhang et al., 2019) and phonological features (Staib et al., 2020).

Additionally, various modeling schemes have been developed to encourage knowledge sharing between languages, such as parameter generation networks (Nekvinda and Dusek, 2020) and leveraging unpaired speech or text data for pre-training (Saeki et al., 2023b,a). Despite these efforts, most prior work still covers a small number of languages but there are a few efforts which scaled to 46 languages (He et al., 2021) or use unsupervised techniques to scale to 101 languages (Saeki et al., 2023b). Meyer et al. (2022) also builds VITS models based on readings of the Bible but their work is limited to ten African languages.

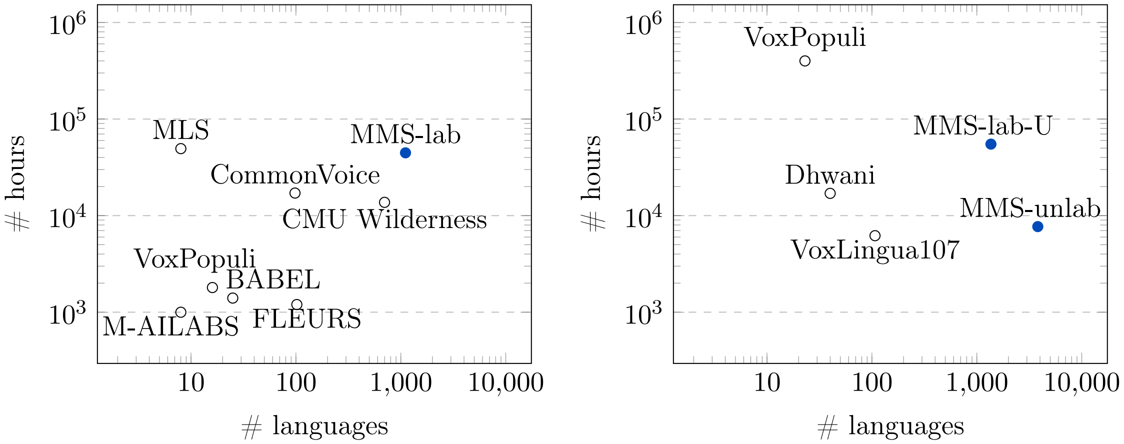
Other Speech Processing Tasks. Speech processing is a broad field and there are many other tasks than ASR, LID and TTS which would benefit from more multilingual systems. However, most efforts on evaluating multilingual systems are focused on ASR or LID (Shi et al., 2023). The Zero Resource Speech Challenge provides a range of interesting tasks such as ABX discrimination but these tasks are only available for English (Dunbar et al., 2021). Query-by-example (QbE) speech search is the task of retrieving utterances relevant to a given spoken query (Kamper et al., 2019) and it could be interesting to make this task available in many more languages. There is also work on multilingual keyword spotting which is the task of identifying whether a particular word was spoken in an utterance (Mazumder et al., 2021).

Multilingual NLP. Multilinguality has been a very active research area in NLP where researchers introduced cross-lingually pre-trained sentence encoders (Conneau and Lample, 2019) spanning 100 languages, or pre-trained multilingual sequence to sequence models (Liu et al., 2020) applied to machine translation. Multilingual machine translation models have also been scaled to 200 languages (NLLB Team et al., 2022) and even 1,000 languages (Bapna et al., 2022a).

3. Dataset Creation

Our work leverages two new datasets to expand the language coverage of speech technology. In this section, we first detail how we create a labeled dataset which includes speech audio paired with corresponding text in 1,107 languages (MMS-lab; 44.7K hours; §3.1). Second, we discuss the creation of an unlabeled dataset for which we only have audio recordings and no corresponding text. This dataset spans 3,809 languages (MMS-unlab; 7.7K total hours; §3.2).

We also use an unlabeled version of MMS-lab for pre-training and language identification. This spans a larger number of languages, as we can also use unlabeled audio from our data source (MMS-lab-U; 1,362 languages; 55K hours). Figure 2 compares the datasets to existing corpora. A full list of the languages supported is available at `https://github.com/facebookresearch/fairseq/tree/main/examples/mms`.



(a) Labeled Datasets.

(b) Unlabeled Datasets.

Figure 2: **Dataset Overview.** MMS-lab, MMS-lab-U and MMS-unlab compared to existing multilingual speech corpora in terms of the supported languages and dataset size. We compare to BABEL (Gales et al., 2014), CMU Wilderness (Black, 2019), CommonVoice (Ardila et al., 2020), Dhwani (Javed et al., 2022), FLEURS (Conneau et al., 2022), M-AILABS (M-AILABS, 2018), MLS (Pratap et al., 2020c), VoxLingua107 (Valk and Alumäe, 2020), and VoxPopuli (Wang et al., 2021) .

3.1 Paired Data for 1,107 Languages (MMS-lab)

We obtain speech data and transcriptions for 1,107 languages by aligning New Testament texts obtained from online sources (§3.1.1) using the following steps:

1. Download and preprocess both the speech audio and the text data (§3.1.2).

2. Apply a scalable alignment algorithm which can force align very long audio files with text and do this for data in 1000+ languages (§3.1.3).

3. Initial Data Alignment: we train an initial alignment model using existing multilingual speech datasets covering 8K hours of data in 127 languages and use this model to align data for all languages (§3.1.4).

4. Improved Data Alignment: we train a second alignment model on the newly aligned data for which the original alignment model has high confidence and generate the alignments again. The new alignment model supports 1,130 languages and 31K hours of data including the data used in step 3 (§3.1.5).

5. Final data filtering: we filter the low-quality samples of each language based on a cross-validation procedure. For each language, we train a monolingual ASR model on half of the aligned data to transcribe the other half of the data. We retain only samples for which the transcriptions are of acceptable quality (§3.1.6).

6. We partition the data into training, development and test portions (§3.1.7).

3.1.1 Data Source

The MMS-lab dataset is based on recordings of people reading the New Testament in different languages. The New Testament consists of 27 books and a total of 260 chapters. Each chapter is divided into a number of verses which can be between one and several sentences long. Specifically, we obtain data from Faith Comes By Hearing[1], goto.bible and bible.com. This includes the original text data as we well as the corresponding audio recording.

Basic Data Characteristics. The data sources provide 1,626 audio recordings of the New Testament in 1,362 languages, totaling 55K hours and we refer to this data as the MMS-lab-U dataset.[2] Out of these, both text and audio is available for 1,306 different recordings in 1,130 languages and a total of 49K hours, which we focus on for MMS-lab. For 99 languages, we have multiple recordings. Each recording provides separate audio files for each chapter and the duration of each chapter is on average 6.7 minutes but there is significant variance, depending on the language and chapter. Recordings are almost always single speaker which makes it well suited for building speech synthesis systems (§7). However, speakers are often male which may introduce unwanted biases into machine learning models which we analyze below (§8.1).

Multiple Scripts or Dialects per Language. When there are multiple recordings per language, we found that some recordings differ in the used writing script, e.g., for Serbian there are recordings using the Latin script and another which uses Cyrillic.[3] Recordings can also differ in the spoken dialect, e.g., for the Poqomchi' language there are recordings with western and eastern dialects.

Depending on the downstream task, we handle these cases differently: for language identification, we merge the different recordings regardless of writing script or dialect, for speech synthesis models, we choose one recording per dialect/script to avoid introducing additional speakers into the training data, and for automatic speech recognition, we combine all the recordings within the same script/dialect into one and treat them as different languages, e.g., srp-script:latin, srp-script:cyrillic.

Micro and Macro Language Distinction. When there is a micro and macro language distinction, then we generally keep micro languages as distinct languages. However, for languages which are part of benchmarks we use for evaluation, e.g., FLEURS (Conneau et al., 2022), we deviate from this policy if the respective benchmark only contains the macro language, by merging the micro languages. For example, Azerbaijani is a macro language and MMS-lab provides data for two associated micro languages, North Azerbaijani and South Azerbaijani. For ASR, our evaluation benchmarks also keep the same distinction and so we model both micro languages separately, however, for LID, one of our benchmarks, VoxLingua-107 (Valk and Alumäe, 2020), only contains the macro language, and therefore, for LID only, we merge both micro languages into a single macro language.

Recordings with Background Music. Some of the recordings contain background music and we refer to these as drama recordings. In our final MMS-lab dataset, 38% of languages

[1] `https://www.faithcomesbyhearing.com/`

[2] For the audio files with no paired text, we perform VAD and segment them into smaller files.

[3] Specifically, we measure the correlation of the character frequency distribution for all pairs of the recordings and examine recordings with a correlation lower than 0.99 further.

are represented solely by a drama recording and 11% have both drama recordings and recordings without background music (non-drama recordings). For speech synthesis, we apply pre-processing to remove background music (§7).

While this data source covers a lot of languages, it requires careful curation to make it usable for building high-quality models which presents unique challenges given the large number of languages. We detail the steps we take in the remainder of this section.

Potential Biases. The MMS-lab data covers many languages but it also has potential for biases that we would like to acknowledge: the data is from narrow and biased domain (religion) and most recordings are from a single speaker that is almost always male. This poses the risk of transcriptions and other biases towards religion as well as poor performance on non-male speakers. We analyze some of the biases in more detail in §8 and find that models trained on the data show only small biases.

3.1.2 Data Pre-processing

Speech. The original audio files are available in MP3 stereo format using a 22/24/44 kHz sampling rate. We convert all of them to a single channel and 16 kHz sampling rate.

Text Normalization. We design a generic text normalization pipeline that works well across the languages we consider. First, we perform Unicode NFKC normalization[4] and lower case all characters. NFKC normalization helps to make sure the character encoding is consistent. Next, we remove HTML tags such as "&gt;" or "nbsp;". We also remove punctuation and try to perform this carefully by including characters for which we are confident that they are in fact punctuation.[5]

We noticed that some recordings have a relatively high rate of brackets in the text: our criteria is recordings where at least 3% of verses contain brackets which resulted in about 50 recordings. We listened to a few instances of each recording to verify whether the text in the brackets is present in the audio or not. In many cases, we noticed that the text in the brackets was not spoken and we removed the brackets and the text within.

3.1.3 Scalable Forced Alignment

The chapter recordings from the data source can be up to 43 minutes long which cannot be directly used by current machine learning algorithms: we use Transformer models which require large amounts of GPU memory and their computational complexity is quadratic in the input size. We therefore segment the data into smaller units so it can be used by standard algorithms. Forced alignment determines which parts of the audio correspond to which parts of the text.

Given this alignment, the data can be segmented in different ways. In our case, we segment the data into individual verses which are typically a single sentence but can sometimes contain several sentences. The average duration of a verse is about 12 seconds. Figure 3 illustrates how the alignments enable creating verse-level audio segments for each chapter recording. In this section we detail how we perform efficient forced alignment on GPUs.

[4] For example, there are two ways to represent Ç (Latin C with combining cedilla): splitting it into Latin capital C and combining cedilla (NFKD), or having a single Unicode character C with cedilla (NFKC).

[5] We obtain an initial set of punctuation characters from the respective Unicode category.

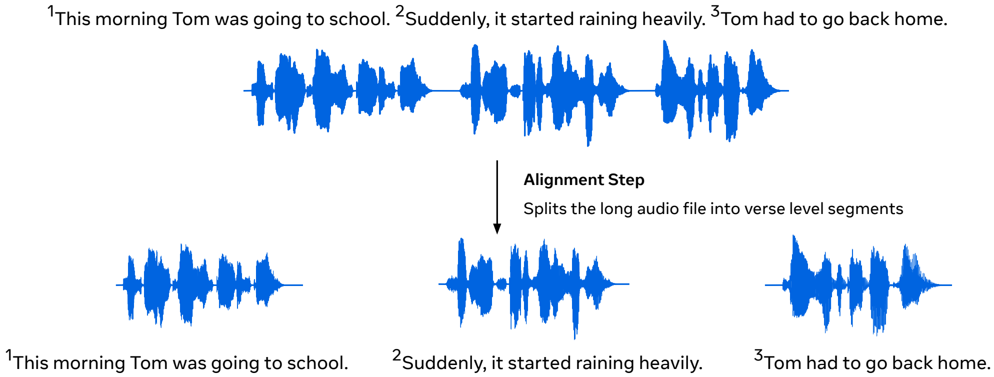

[1]This morning Tom was going to school. [2]Suddenly, it started raining heavily. [3]Tom had to go back home.
Alignment Step
Splits the long audio file into verse level segments
[1]This morning Tom was going to school.
[2]Suddenly, it started raining heavily.
[3]Tom had to go back home.

Figure 3: **Illustration of Data Alignment.** Forced alignment enables segmenting audio recordings and corresponding text into smaller segments that can be used to train machine learning models.

Generating Posterior Probabilities. Forced alignment requires posterior probabilities from an acoustic model which we use for alignment (§3.1.4). This acoustic model is a Transformer which requires substantial amounts of memory to store activations which makes it infeasible to use for long audio files. As a workaround, we chunk the audio files into 15 second segments, generate posterior probabilities for each audio frame using the alignment model, and then concatenate these posterior probabilities into a single matrix again. The acoustic model is trained with Connectionist Temporal Classification (CTC; Graves et al. 2006).

Forced Alignment using CTC. Next, we perform forced alignment which finds the most likely path in the posterior probabilities for a given input audio sequence of length T and a text transcription of length L. These posterior probabilities require $\mathcal{O}(T \times L)$ memory and a path will be of length T. This path is computed using the Viterbi algorithm. There are open source libraries implementing the algorithm on CPU (Kürzinger et al., 2020; Kahn et al., 2022), however, the CPU versions are slow to run, particularly on long recordings, as we will show below.

Efficient Forced Alignment on GPUs. In order to make force alignment efficient for our purpose, we implemented a GPU version that computes the Viterbi path memory in a memory efficient way. Storing all $\mathcal{O}(T \times L)$ forward values for the Viterbi algorithm is infeasible on GPUs due to memory constraints. We therefore only store forward values for the current and the previous time-step and regularly transfer the computed backtracking matrices to CPU memory. This reduces the required GPU memory to $\mathcal{O}(L)$ compared to $\mathcal{O}(T \times L)$ and enables forced alignment for very long audio sequences at high speed. Appendix A illustrates the algorithm and an implementation is available as part of TorchAudio (Yang et al., 2021).[6]

[6] `https://github.com/pytorch/audio`

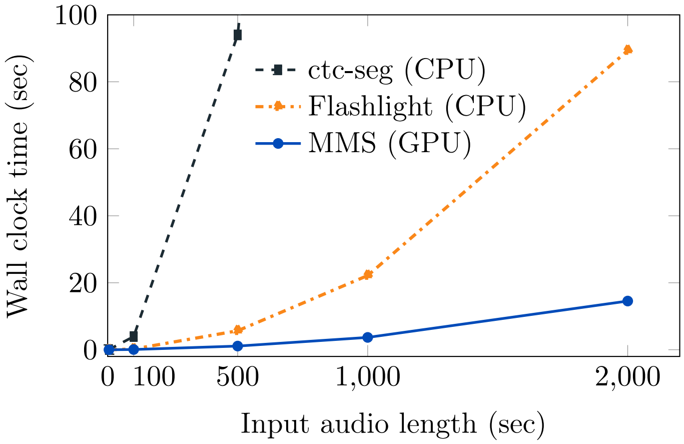


Figure 4: **Efficiency of Forced Alignment Implementations.** The MMS implementation runs on GPU and can process long audio sequences in reasonable time compared to CPU alternatives.

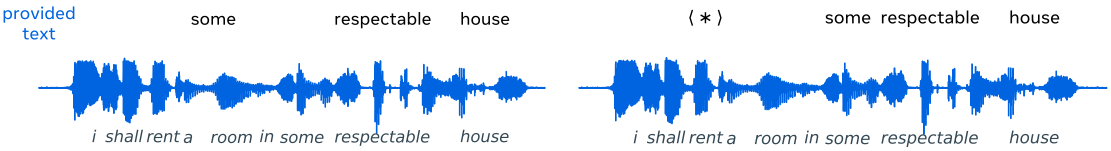
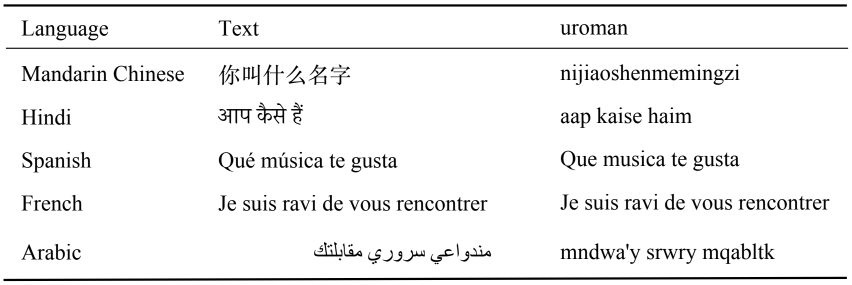


Figure 5: **Illustration of the ⟨∗⟩ Token in Forced Alignment.** We show the text to which we would like to align the audio at the top and what is actually spoken at the bottom in italics. Left: erroneous alignment when the provided text is incomplete. The word *some* is aligned incorrectly. Right: using ⟨∗⟩ token at the beginning enables correct alignment of the provided text.

Figure 4 shows that the forced alignment implementation scales much better to longer sequences than CPU alternatives such as ctc-segmentation (Kürzinger et al., 2020), a popular segmentation library used in ESPNet (Watanabe et al., 2018), SpeechBrain (Ravanelli et al., 2021) and Flashlight (Kahn et al., 2022).

Robust Alignment for Noisy Transcripts. For many recordings, speakers introduce the chapter name and the version of the New Testament before reading the first verse, however, the corresponding text does not contain this information. This is problematic for forced alignment because the algorithm will still try to align the beginning of the audio to the text which can result in incorrect alignments. Another challenge is that numbers are generally spelled as digits in the text whereas our alignment model is trained on existing corpora which follow common practice of spelling numbers out fully (§3.1.4). Spelling numbers out requires language-specific and hand-crafted tooling which is not available for the 1,107 languages we consider.

To enable robust alignment in both cases, we introduce a star token (⟨∗⟩; Pratap et al. 2022; Cai et al. 2022) to which audio segments can be mapped if there is no good alternative



Language	Text	uroman
Mandarin Chinese	你叫什么名字	nijiaoshenmemingzi
Hindi	आप कैसे हैं	aap kaise haim
Spanish	Qué música te gusta	Que musica te gusta
French	Je suis ravi de vous rencontrer	Je suis ravi de vous rencontrer
Arabic	مندواعي سروري مقابلتك	mndwa'y srwry mqabltk

Table 1: **Illustration of Text Encoding for Forced Alignment.** Example outputs of uroman (Hermjakob et al., 2018).

in the text.[7] We insert ⟨∗⟩ at the beginning of the text data of each chapter and replace numerical digits with ⟨∗⟩ throughout. The posterior probability for this token is set to one. After alignment, we add back the original digits and the subsequent data filtering often removes segments where the audio contains additional information not present in the aligned text (§3.1.6). Figure 5 illustrates how adding the ⟨∗⟩ token helps to improve alignment when the paired text does not cover the beginning of the audio.

3.1.4 Initial Data Alignment

Acoustic Model. To perform the forced alignment, prior work typically uses acoustic models trained on data in the same language. For example, for the eight languages of the MLS dataset Pratap et al. (2020c) the authors used acoustic models trained on existing data for these languages. However, our setting includes many languages for which no datasets or acoustic models exist. We therefore train a multilingual acoustic model on FLEURS (Conneau et al., 2022) and CommonVoice 8.0 (Ardila et al., 2020) to learn a shared representation across languages which we hope to generalize to unseen languages. The multilingual model is based on fine-tuning XLS-R (Babu et al., 2022) using a total 8K hours of data covering 127 languages.

Text Encoding. The text data is represented using the uroman transliteration tool (Hermjakob et al., 2018) which maps different writing scripts to a common Latin script representation.[8] This is done using character descriptions based on Unicode tables and a large number of additional heuristics and it has language-specific romanization rules for some languages. Prior work (Black, 2019) used Unitran (Yoon et al., 2007; Qian et al., 2010; Black, 2019) which converts UTF-8 encoded text into a phonetic transcription in either WorldBet (Hieronymus, 1993) or X-SAMPA (Wells, 1995).

Inspired by Black (2019), we initially investigated X-SAMPA but found that uroman led to similar quality results and we decided to use uroman since it is easier to interpret compared to International Phonetic Alphabet (IPA) based symbols generated by Unitran. Table 1 shows some example outputs from uroman. We lowercase all the letters of the

[7] This is different from an OOV token or silence token in an HMM topology, which model a certain type of acoustic behavior and are independent from other in-vocabulary tokens.

[8] `https://www.isi.edu/~ulf/uroman.html`

uroman output and retain only *a* to *z* characters as well as the apostrophe character to train acoustic models for forced alignment (§ 3.1.3).

3.1.5 Improved Data Alignment

To improve the alignments, we use a subset of good-quality samples to train a new alignment model. Samples are selected based on a score which is the length-normalized difference between the probability of the forced alignment path $P(Y^{aligned} \mid X)$ and the probability of greedy decoding from the alignment model $P(Y^{greedy} \mid X)$. The former is constrained by the text used in the alignment and the latter is unconstrained. A large difference between these two quantities may indicate that the alignment is incorrect.[9] Concretely, the score is

$$\frac{1}{T} \log P(Y^{aligned} \mid X) - \log P(Y^{greedy} \mid X) \tag{1}$$

where T is the length of the audio. The score can vary from $-\inf$, indicating low quality samples, to 0, indicating high quality samples. After manual inspection of sample quality and their corresponding score for several languages, we select -0.2 as the threshold and choose samples with scores greater than this threshold. The improved alignment model is trained on a total of 31K hours in 1,130 languages which includes the data we used for the initial alignment model.[10] We use this new model to re-generate verse level alignments for our data.

3.1.6 Final Data Filtering

After generating the improved alignments, we noticed that some samples are still of low quality. Some recordings are not entirely faithful to the text and speakers sometimes add their own interpretation or paraphrase parts of the text. This will negatively impact ASR and TTS models trained on the data and we therefore perform a final data filtering step to improve data quality as much as possible.

We we train monolingual ASR models on half of the the aligned samples of each recording, measure performance on the remaining half and remove samples which have a character error rate (CER) in excess of 10%. This removes about 1.7% of all samples across all languages.

High CER may be due to low quality samples in either half of the data, that is the training data of the ASR model or the data being evaluated. To help ensure we use recordings that are generally of good quality, we remove 3837 recordings which have CER in excess of 5% on the development set. We retain a total of 1,239 recordings covering 1,107 languages.

3.1.7 Creating train/dev/test splits

The recordings often contain only a single speaker which makes it challenging to measure generalization performance of models trained on this data. This is why we evaluate models trained on MMS-lab data on existing benchmarks as much as possible in the remainder of this paper. The advantage of this is that it side steps the aforementioned issues and it also enables us to better understand how the data can be useful in other domains.

[9] It may also indicate that the alignment model is of poor quality but in practice we found that the metric identifies many incorrect alignments.

[10] This model is available at `https://github.com/facebookresearch/fairseq/tree/main/examples/mms`

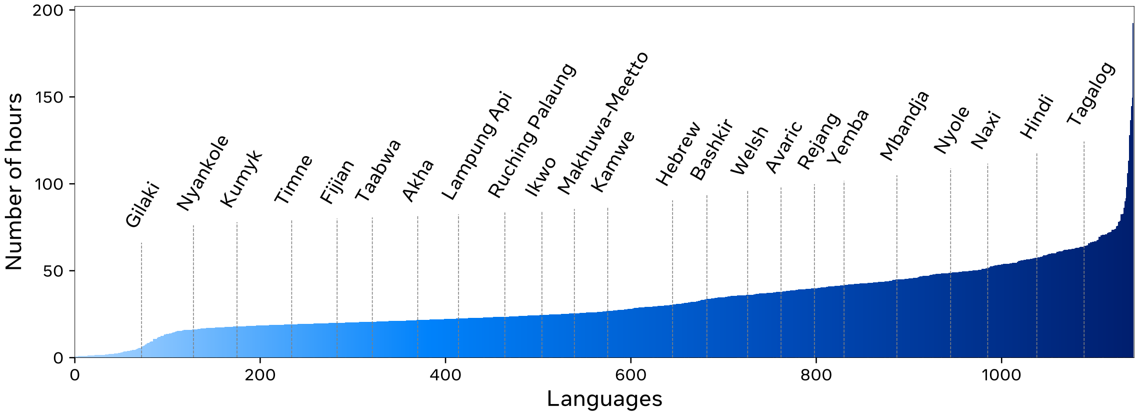


Figure 6: **MMS-lab: Amount of Speech Data across Languages.** We show the size of the training data sets and name some of the 1,107 languages.

However, existing benchmarks only cover a small fraction of the languages in MMS-lab and in order to be able to develop models for all languages, we split the aligned recordings into train/dev/test splits. We aim to make the content of the splits as disjoint from each other as possible and use a similar split across different recordings. To do so, we try to use different books for each split and use the same books for each split across recordings and languages as much as possible.

Concretely, we use the book Mark (MRK) as development set, the book John (JHN) as test set and the remaining books for training. For the 147 recordings where not all 260 chapters are available, we deviate from this and make a best effort split by books depending on which books are available in the respective recording. In this case, we aim to have at least 10% of all available data in the development set and the test set each, or at most two hours of data in each set, whichever is less.

The final dataset contains 44.7K hours of paired speech data where we use 36.8K hours for training (82.3%), 3.5K hours for development (7.8%) and 4.4K hours for testing (9.9%). For each language, the train split contains an average of 32 hours (stddev=19), the dev split contains an average of 3.1 hours (stddev=1.8) and test split an average of 3.9 hours (stddev=2.3). Figure 6 shows the data distribution across languages.

3.2 Unpaired Data for 3,809 Languages (MMS-unlab)

Data Source. The data source for this dataset is Global Recordings Network which provides recordings of Bible stories, evangelistic messages, scripture readings, and songs in more than 6,255 languages and dialects.[11] The audio files are not accompanied by a corresponding text transcription but the source makes clear which language is being spoken. We group the data by language, combining dialects of the same language resulting in a total of 3,860 languages and 9,345 hours of audio.

[11] `https://globalrecordings.net/`

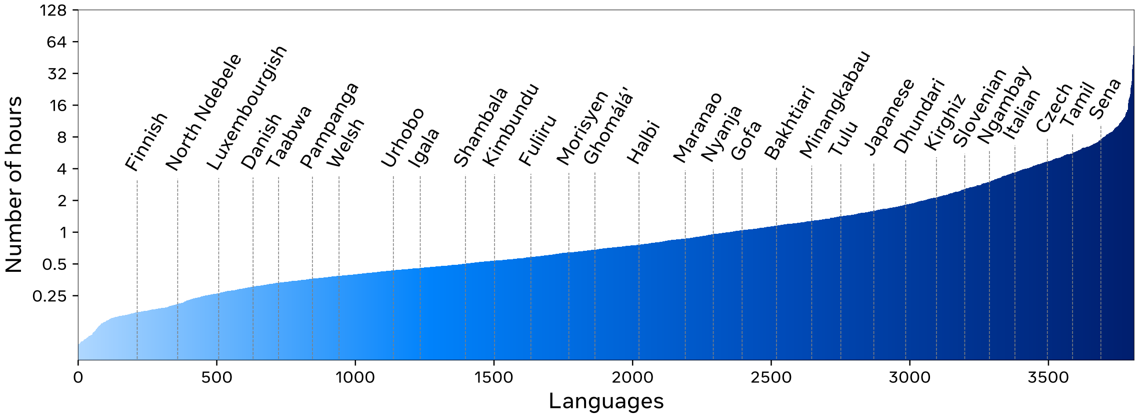


Figure 7: **MMS-unlab: Amount of Speech Data across Languages.** We show the size of the training data sets and name a few of the 3,809 languages.

Pre-processing. We convert the audio files to single channel and a sample rate of 16kHz. Next, we use inaSpeechSegmenter (Doukhan et al., 2018), a CNN-based audio segmentation model, to identify segments of speech, music, noise and silence in the audio. If two segments of speech are separated by intermediate segments containing music or noise, then we consider joining these segments if the intermediate segment is no longer than 20% of all segments together. This is to build samples that are of longer duration and still contain mostly speech. The remaining non-speech segments are discarded.

Next, we randomly split the speech segments into portions of between 5.5 and 30 seconds. This makes the data usable for training downstream models and creates a length distribution of the samples that is in line with other datasets such as FLEURS where the average sample length is about 12 seconds.

Dataset Split. Finally, we split the samples of each language randomly into 80% training data, 10% development data, and 10% test data. We also remove 51 languages for which we have less than 5 minutes of training data to ensure we have sufficient data to train models. The final dataset comprises a total of 7.7K hours of data in 3,809 languages. The training portion is 6.2K hours and there are 770 hours for the valid and test sets each. For each language, the train set contains an average of 97 minutes (stddev=177.4), and the dev/test sets contain an average of 12.1 minutes (stddev=22.3). Figure 7 shows the data distribution across languages.

3.3 Comparison to Existing Broad Coverage Approaches and Other Datasets

In this section, we present a comparison to two related studies which aimed to expand speech technology to many languages. The CMU Wilderness project (Black, 2019) also used New Testament data to build speech synthesis models for 699 languages (§3.3.1) and ASR-2K (Li et al., 2022) focused on automatic speech recognition for nearly two thousand languages (§3.3.2). Finally, we assess the viability of the new data for building machine learning models by comparing the performance of ASR models trained on MMS-lab to models trained on an existing dataset in an out-of-domain setting (§3.3.3).

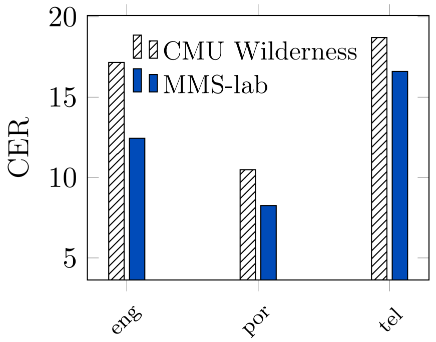


Figure 8: **MMS-lab vs. CMU Wilderness.** Character Error Rate of ASR models in English (eng), Portuguese (por) and Telugu (tel) on the FLEURS dev set.

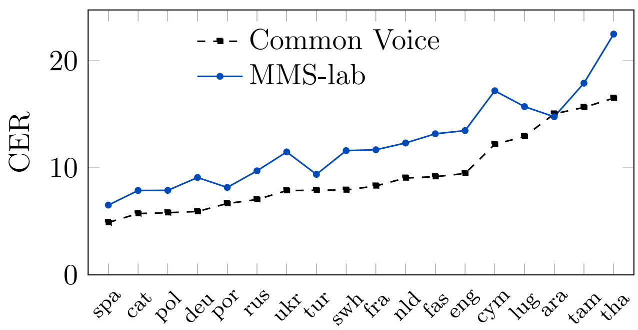


Figure 9: **MMS-lab vs. Common Voice.** Character Error Rate on FLEURS dev set for ASR models trained on Common Voice (CV) data and MMS-lab data for 18 languages.

3.3.1 CMU Wilderness Dataset

The most comparable prior work is the CMU Wilderness project which used data from similar sources (Black, 2019). To better understand how our data creation process compares to their method, we conduct a best effort side-by-side reproduction of their process and use the resulting data to train monolingual ASR models under the same settings.[12]

To compare the effectiveness of their data creation method to ours (§3.1), we take the original data from our data source and apply either our protocol or the protocol of Black (2019). For languages where multiple recordings exist, we only use the recordings used in the CMU Wilderness dataset to enable a better comparison. Next, we use the resulting data to fine-tune XLS-R models (Babu et al., 2022) for monolingual ASR and then measure accuracy in terms of character error rate on the FLEURS dev set.

Figure 8 shows that the MMS-lab data preparation process results in better quality ASR models compared to CMU Wilderness with improvements between 2.1%-4.7% CER, depending on the language. Our alignment procedure also retains a much larger amount of the training data compared to the CMU Wilderness protocol: for Telugu, there are 26.5 hours of data, MMS-lab retains 26.2 hours compared to 11.1 hours for the CMU Wilderness process. For English, we start with 17.3 hours, MMS-lab retains 17 hours vs. 10.6 hours for CMU Wilderness.

3.3.2 ASR-2K

The most comparable multilingual ASR work to ours is ASR-2K (Li et al., 2022) which covers 1,909 languages. Their approach is based on mapping the output of an eight language multilingual model to appropriate phonemes for the language of interest. In contrast, MMS-lab has actual paired speech and text data.

[12] We reproduced the data by following the steps outlined at `https://github.com/festvox/datasets-CMU_Wilderness`

ASR-2K reports average character error rate (CER) on 34 languages of Common Voice 6.0 and uses a language model for decoding. MMS-lab covers 22 out of these 34 languages and we use it to train monolingual ASR models without language models. The monolingual models trained on MMS-lab dataset obtain an average CER of 9.6 on 22 languages. ASR-2K reports CER 50.9 on 34 languages. While not a like for like comparison, this difference suggest that MMS-lab enables higher quality ASR models. We stress that this is a best effort comparison and does not enable strong conclusions.[13]

3.3.3 Other Existing Datasets

The MMS-lab dataset covers a large number of languages but it also has potential downsides: it is both from a particular narrow domain and most recordings are from a single speaker. This may lead to poor performance on other domains or when the systems are applied to unseen speakers.

To get a better sense of both issues, we train monolingual ASR models by finetuning XLSR (Babu et al., 2022) on MMS-lab and evaluate these models on the FLEURS benchmark. For comparison, we train another set of ASR models on labeled data from Common Voice which is an existing dataset and does not have the aforementioned downsides: the domain is general and the data contains multiple speakers. We control the amount of training data of both datasets by using exactly ten hours of training data and one hour of development data from both datasets.

Figure 9 shows that models trained on CommonVoice perform better on 18 languages of FLEURS (average CER 9.3 vs. 12.2) but the models MMS-lab still enable good performance.[14] This is despite the fact that MMS-lab utterances are often from a single speaker and are from a very narrow domain. While there is certainly higher quality data for head languages, this result suggests that the quality of the MMS-lab data can enable high quality speech systems for a large number of other languages.

4. Cross-lingual Self-supervised Speech Representation Learning

As a first step, we train a self-supervised model of speech representations(Baevski et al., 2020b; Hsu et al., 2021b; Chen et al., 2022a) on the data outlined above as well as other existing public corpora. We use wav2vec 2.0 for pre-training on unlabeled data which we later use as the basis for several downstream speech tasks (Baevski et al. 2020b; §4.1). The resulting models were pre-trained on 1,406 languages which is over four times the number of languages of known prior work (Zhang et al. 2023a; §4.2). The increased language coverage results in better performance for both ASR and LID compared to XLS-R (Babu et al. 2022; §4.3) which covered 128 languages and is publicly available.

[13] We could not obtain a per-language break down of the ASR-2K results.

[14] Spanish (spa), Catalan (cat), Polish (pol), German (deu), Portoguese (por), Russian (rus), Ukrainian (ukr), Turkish (tur), Swahili (swh), French (fra), Dutch (nld), Farsi (fas), English (eng), Welsh (cym), Luganda (lug), Arabic (ara), Tamil (tam), Thai (tha).

4.1 Method: wav2vec 2.0 and XLS-R

Our work builds on Babu et al. (2022) who pretrain wav2vec models on data from multiple languages. The wav2vec project created a series of models for learning self-supervised speech representations (Schneider et al., 2019; Baevski et al., 2020a,b) from unlabeled speech data. The resulting models can then be used to solve downstream speech tasks by fine-tuning them on labeled data or by tackling these tasks without labeled data using unsupervised learning (Baevski et al., 2021; Liu et al., 2022).

The most prominent one is wav2vec 2.0 (Baevski et al., 2020b) which enables building speech recognition models with only ten minutes of labeled data and even no labeled data at all (Baevski et al., 2021). The basic architecture of wav2vec 2.0 is as follows: a convolutional feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ maps raw audio $\mathcal{X}$ to latent speech representations $z_1, \ldots, z_T$ which are input to a Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ to output context representations $c_1, \ldots, c_T$ (Baevski et al., 2020a). Each z_t represents 25ms of audio strided by 20ms and the Transformer architecture follows BERT (Vaswani et al., 2017; Devlin et al., 2019).

During training the feature encoder representations are discretized to $q_1, \ldots, q_T$ with a quantization module $\mathcal{Z} \mapsto \mathcal{Q}$ to represent the targets in the objective. The quantization module uses a Gumbel softmax to choose entries from the codebooks and the chosen entries are concatenated (Jegou et al., 2011; Jang et al., 2016; Baevski et al., 2020a).

The model is trained by solving a contrastive task over masked feature encoder outputs. At training time, spans of ten time steps with random starting indices are masked. The objective requires identifying the true quantized latent q_t for a masked time-step within a set of $K = 100$ distractors sampled from other masked time steps. The objective is augmented by a codebook diversity penalty to encourage the model to use all codebook entries (Dieleman et al., 2018).

XLSR and XLS-R train wav2vec 2.0 on many different languages from several datasets to obtain cross-lingual representations (Conneau et al., 2020a; Babu et al., 2022). In order to balance the training data, two data sampling steps are performed. First, for each dataset, we sample the data for the different languages L from a distribution $p_l \sim \left(\frac{n_l}{N}\right)^{\beta_L}$ where $l = 1, \ldots, L$, n_l is the amount of unlabeled data for each language in the dataset, N is the total amount of training in the dataset, and β_L is the upsampling factor which controls the trade-off between high- and low-resource languages during pretraining. Second, we balance the different datasets by treating each dataset as a language in the above sampling scheme with a sampling parameter β_D.

4.2 Pre-training Setup

Hyperparameters. We largely follow prior work in training cross-lingual wav2vec 2.0 models (Conneau et al., 2020a; Babu et al., 2022) and use the wav2vec 2.0 implementation available in fairseq (Ott et al., 2019) to train models with roughly 300M and 1B parameters (Table 2). To make efficient use of GPU memory, we use a fully sharded backend (Rajbhandari et al., 2021) as well as activation checkpointing (Chen et al., 2016) implemented in FairScale (Baines et al., 2021).

Our models are optimized with Adam (Kingma and Ba, 2015) and the learning rate is warmed up for the first 32K steps followed by polynomial decay to zero for the remainder of training. Training audio sequences are cropped to a maximum of 320K samples, or 20

Model	#langs	Datasets	B	M	F	A	#params
Prior work							
XLSR-53	53	MLS, CV, BBL	24	1024	4096	16	317M
VP-100K	23	VP-100K	24	1024	4096	16	317M
XLS-R (0.3B)	128	VP-400K, MLS, CV, VL, BBL	24	1024	4096	16	317M
XLS-R (1B)	128	VP-400K, MLS, CV, VL, BBL	48	1024	4096	16	965M
MMS (0.3B)	1,406	MMS-lab, VP-400K, MLS, CV, VL, BBL	24	1024	4096	16	317M
MMS (1B)	1,406	MMS-lab, VP-400K, MLS, CV, VL, BBL	48	1024	4096	16	965M

Table 2: **Self-supervised Models.** Details of our models including prior work: XLSR-53 (Conneau et al., 2020a), VP-100K (Wang et al., 2021), XLS-R (Babu et al., 2022) and the MMS models: the number of languages (#langs), pretraining data (Datasets), the number of Transformer blocks (B), the number of hidden states (M), the inner dimension of feed-forward blocks (F), the number of attention heads (A) and the total number of parameters (#params).

seconds, and all models were pre-trained for a total of one million updates on A100 GPUs with 80GB of memory. The MMS (0.3B) model was trained with an effective batch size of 2.3 hours of data across 48 GPUs and the MMS (1B) model was trained with an effective batch size of 3.5 hours on 64 GPUs.

Data. The pre-training data covers about 491K hours in 1,406 languages. This data is drawn from six training corpora with different characteristics, including the corpora used in XLS-R (Babu et al., 2022):

- MMS-lab-U: 1,362 languages comprising 55K hours (§3.1).
- Multilingual Librispech (MLS): 8 European languages of read books totaling 50K hours (Pratap et al., 2020c)
- CommonVoice (CV): 89 languages totaling 8.8 hours of read text; we use v9.0 of the corpus (Ardila et al., 2020))
- VoxLingua-107 (VL): 107 languages totaling 5.3K hours of YouTube content (Valk and Alumäe, 2020)
- BABEL (BBL): 17 African and Asian languages totaling about 1K hours of conversational telephone data (Gales et al., 2014)
- VoxPopuli (VP): 371K hours of unlabeled speech data in 23 languages derived from European Parliament event recordings (Wang et al., 2021)

Pre-training only uses the speech audio and none of the transcriptions and we balance the data following the strategy outlined in §4.1 using $\beta_L = \beta_D = 0.5$

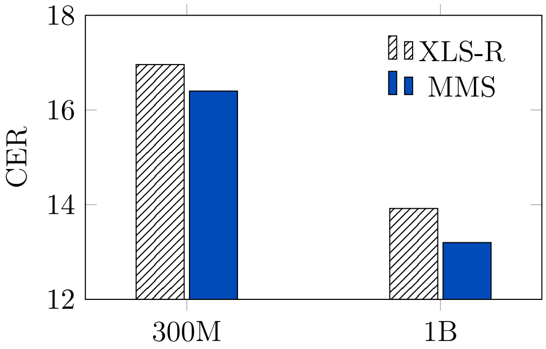


Figure 10: **MMS vs. XLS-R.** Character error rate (CER) on 61 FLEURS languages when fine-tuning multilingual ASR models on MMS-lab data. We report average performance on FLEURS dev data.

4.3 Comparison to XLS-R

To better understand how the MMS models compare to XLS-R we fine-tuned both for automatic speech recognition on the 61 languages of the FLEURS benchmark for which MMS-lab provides training data. Models are evaluated without a language model and we report the average character error rate (CER) on FLEURS development data over all languages.

Figure 10 shows that the MMS models perform better: for the 300M size, MMS has 0.6 lower CER than XLS-R, and for the 1B size, the difference is 0.7 CER. More capacity helps to improve performance: for XLS-R the error rate decreases by 3.2 CER absolute when scaling the number of parameters from 300M to 1B and for MMS-lab there is a 3.0 CER improvement.

MMS pre-trains on over ten times the number of languages of XLS-R and this improves performance, particularly on low resource languages (Figure 11) such as Amharic (amh), Lao (lao) or Malayalam (mal). Compared to XLS-R, the pre-training data of MMS covers the following languages which improve: Chewa (nya), Fulah (ful) and Oromo (orm). However, improvements at low resource languages result in a small degradation in some of the high-resource languages such as English (eng) or Spanish (spa) but there are also other languages such as Tajik (tgk) or Welsh (cym) which perform less well.

Equipped with this new pre-trained model we now investigate the use of MMS-lab and MMS-unlab for several downstream tasks.

5. Automatic Speech Recognition

We first turn to the task of transcribing speech in up to 1,107 different languages. We use the labeled dataset we collected (§3.1) to fine-tune our pre-trained models (§4) for ASR. We first outline our modeling approach (§5.1) and then scale the number of languages for multilingual ASR from 61 to 1,107 in order to better understand the impact of supporting more languages (§5.2). Next, we compare our models to existing multilingual work (§5.3), and build robust multilingual models supporting 1,162 languages trained on several existing

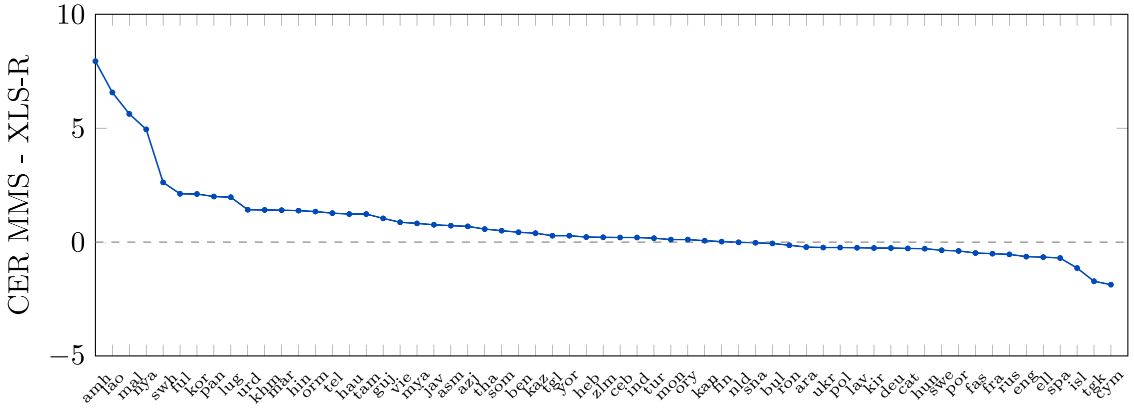


Figure 11: **MMS vs. XLS-R Breakdown.** We show the absolute character error rate difference between multilingual ASR models based on XLS-R and MMS models with 1B parameters. Positive values indicate better performance of MMS and negative values better performance of XLS-R. Models are fine-tuned on MMS-lab data and evaluated on the development sets of 61 FLEURS languages.

corpora as well as the MMS-lab data (§5.4). Finally, we evaluate our multilingual models on all languages they support (§5.5).

5.1 Modeling and Training Approach

We train multilingual speech recognition models by fine-tuning our pre-trained MMS (1B) model (§4) using labeled data, similar to Baevski et al. (2020b). To output transcriptions, we add a linear layer on top of the model which maps to an output vocabulary which is the set of letters in the labeled training data of all languages considered in a particular setting. Next, we fine-tune the entire model with the Connectionist Temporal Classification (CTC) criterion (Graves et al., 2006).

Optimization. We use Adam (Kingma and Ba, 2015) with exponential decay rates $\beta_1 = 0.9$, $\beta_2 = 0.98$ to train model weights using a tri-stage schedule where the learning rate is warmed up for the first 10% of updates, held constant for the next 40% updates, and then decayed in the final 50% updates. We experimented with different learning rates (1×10^{-4}, 7×10^{-4}, 3×10^{-4} 1×10^{-5}, 7×10^{-6}, 3×10^{-6}, 1×10^{-6}) and number of updates (50K, 100K, 200K, 300K). Unless otherwise mentioned, we fine-tune models for a total of 50K updates with a batch size of 0.8 hours of data using 16 A100 GPUs with 80GB of memory.

Language-specific Adapters, Head and Fine-Tuning (LSAH). In addition to training dense models which share all parameters across languages, we also consider adding adapter modules (Houlsby et al., 2019) to models where we use a different set of adapter weights for each language. Specifically, we introduce adapters at every block of the transformer, and the adapter is added after the last feed-forward block. The adapter module consists of a LayerNorm Ba et al. (2016) layer, a downward linear projection followed by a ReLU activation and an upward linear projection; the inner dimension of the projections is 16.

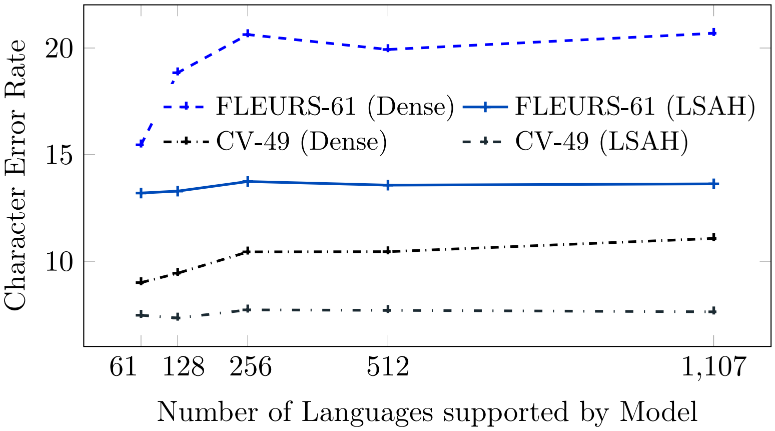


Figure 12: **Scaling Multilingual ASR to 1,107 Languages.** We fine-tune both dense and LSAH models with 61, 128, 256, 512 and 1,107 languages on MMS-lab data and show average CER on 61 languages of FLEURS and 49 languages of CommonVoice. LSAH models have language-specific adapter modules and output layers.

For each language, using an adapter increases the number of parameters by 2M or about 2% of the total number of parameters without adapters. We then perform a second stage of fine-tuning for each language where we introduce a randomly initialized linear layer mapping to the specific output vocabulary of a language in addition to language-specific adapter and fine-tune these additional parameters only for another 2K updates on the labeled data of the respective language.

5.2 Scaling Multilingual ASR to 1,107 Languages

We first analyze training multilingual ASR models with an increasing number of languages by scaling from 61 to 1,107 languages, roughly doubling the number of languages at every step. Models are trained on the labeled data of MMS-lab by fine-tuning the MMS (1B) pre-trained model (§4) and we consider training models with and without language specific adapters and output layers. Each setting is evaluated on the 61 FLEURS languages covered by MMS-lab (FLEURS-61) as well as the 49 languages of CommonVoice covered by MMS-lab (CV-49). Results are reported on the development sets of FLEURS and CommonVoice in terms of Character Error Rate without a language model.

Figure 12 shows that performance degrades quickly for dense models which have no language-specific parameters: for FLEURS-61, CER increases by 5.1 when moving from 61 to 1,107 languages and for CV-49, there is a 2.1 CER increase. This is mainly due to languages being confused with each other which results in large performance drops for certain languages. Language-specific parameters (LSAH) alleviate this issue and show only very little degradation (0.4 CER for FLEURS-61 and 0.2 CER for CV-49).

In summary, this shows that scaling multilingual ASR models to over one thousand languages is feasible and that there is little performance degradation when coupled with language-specific parameters.

	#lang	labeled train	FLEURS-54	
		data (h)	dev	test
Prior Work				
Whisper medium	99	680K	-	50.1
Whisper large-v2	99	680K	-	44.3
This Work				
MMS	61	3K	20.9	20.7
MMS (LSAH)	61	3K	19.0	19.1
MMS	1,107	45K	24.8	24.8
MMS (LSAH)	1,107	45K	18.7	18.7

Table 3: **Comparison to Whisper.** We report average WER on the 54 languages of the FLEURS benchmark supported by both Whisper and MMS (FLEURS-54). MMS is a CTC-based model and to enable a fairer comparison we use n-gram models trained on web data when comparing to Whisper whose decoder is a neural sequence-model that serves as a language model and was trained on billions of web tokens.

5.3 Comparison to Other Work

Next, we present comparisons to other recent related work on multilingual ASR: Whisper (Radford et al., 2022) uses large quantities of labeled web data to train a model supporting 99 languages (§5.3.1) and Google USM (Zhang et al., 2023a) builds multilingual ASR models supporting 100 languages by pre-training on YouTube data (§5.3.2).

5.3.1 Whisper

Whisper is a multilingual model trained on 680K hours of weakly labeled audio data from the web and able to transcribe speech in 99 languages (Radford et al., 2022). The model uses a sequence to sequence architecture (Sutskever et al., 2014) which has a neural sequence model as decoder that acts in part like a language model. The decoder has been trained on the target side text of the labeled training data which likely amounts to several billions of words of text from the web.[15]

In contrast, MMS is a CTC model whose decoder is a simple linear layer mapping to a set of characters (§5.1). When comparing CTC models to models with sequence-model based decoders, the former are typically paired with an external language model to enable a fairer comparison. We therefore train simple n-gram models on web data (Common Crawl) for each language and use it during inference time (CC LM; Heafield 2011; Conneau et al. 2020b; NLLB Team et al. 2022); Appendix B details the training procedure. We evaluate both models on the 54 languages of FLEURS supported by both Whisper and MMS (FLEURS-54) and report word error rate except for Thai, Lao, Burmese and Khmer where we use character error rate.[16]

[15] Assuming 10K words of text per hour of paired speech data, the decoder was trained on about 6.8B words.

[16] We follow Whisper's evaluation methodology but add Khmer to the list of languages where evaluation is in terms of CER because there is no standard tokenization we are aware of.

	FLEURS-102	
	dev	test
Prior Work		
w2v-BERT (Chen et al., 2022b)	-	12.3
Maestro-U (Chen et al., 2022b)	-	8.7
USM (Zhang et al., 2023a)	-	6.9
USM-M (Zhang et al., 2023a)	-	6.5
USM-M-adapter (Zhang et al., 2023a)	-	6.7
This Work		
MMS FL-102 (LSAH) + LM	6.3	6.3

Table 4: **Comparison to Google USM.** We report average CER on the 102 languages of FLEURS and use n-gram models together with our CTC acoustic models. USM-M is an RNN-T model which uses both unlabeled text as well as labeled speech data during pre-training and for MMS we use unlabeled text to train language models for inference.

The results (Table 3) show that MMS reduces the word error rate of Whisper by a relative 58% while supporting over 11 times the number of languages. Moreover, MMS was trained on 44.7K hours of labeled data compared to 680K for Whisper.[17] A reduced version of MMS trained on 61 languages can still outperform Whisper to a similar degree while being trained on only about 3K hours of labeled training data. Overall, MMS (LSAH) outperforms Whisper on 31 out of the 54 languages. Appendix C provides a per-language breakdown of the results.

5.3.2 Google USM

This model is pre-trained on 12M hours of proprietary YouTube audio spanning 300 languages and then fine-tuned to perform ASR for up to 100 languages on a labeled dataset of 90K hours (Zhang et al., 2023a) which results in large improvements over Whisper.[18] In a best effort comparison, we adopt the USM setup where the authors fine-tune their pre-trained model on the labeled FLEURS data. We follow the same training regime as outlined above (§5.1) but fine-tune this model for a total of 300K updates. Results are in terms of character error rate.[19]

There are several important differences between their approach and MMS: first, USM uses an RNN-T model (Chan et al., 2015) which has a built-in neural language model

[17] The non-English portion of the Whisper training data is still 117K hours and covers less than 100 languages, while MMS-lab is less than half the size and covers 11 times the number of languages.

[18] We were not able to obtain the list of languages involved in their comparison to Whisper and therefore resort to a comparison not involving the labeled YouTube data. This has the downside of removing the impact of the labeled datasets of each approach in the comparison.

[19] Zhang et al. (2023a) did not perform any additional pre-processing on the reference transcriptions when reporting CER which enables a comparison to their results [Yu Zhang, personal communication 5 May 2023].

while as MMS is a CTC-based acoustic model.[20] Second, some of the USM models were pre-trained on large quantities of unlabeled text as well as 20K hours of labeled audio data (USM-M/USM-M-adapter) while as MMS is pre-trained only on unlabeled speech data.

To enable a fairer comparison, we use n-gram language models trained on unlabeled text during inference: for each language, we use either an n-gram model trained on CommonCrawl or a model trained on the transcriptions of the FLEURS training set, depending on dev set performance and data availability. Appendix B details the training procedure.

Table 4 shows that MMS performs very competitively compared to USM. We note that the approaches have significant differences in the model architecture and uses of unlabeled/labeled data, however, we believe that the result convinces the reader that a simple CTC model paired with n-gram models can perform very competitively to more advanced architectures and more elaborate pre-training procedures. Appendix D shows a per-language breakdown of the results.

5.4 Robust Multilingual ASR Models

In this section, we turn to building multilingual ASR models on data from multiple domains following a similar approach as prior work for English-only models (Likhomanenko et al., 2020; Chan et al., 2021; Hsu et al., 2021a). We fine-tune the pre-trained MMS (1B) model on MMS-lab, FLEURS, CommonVoice, VoxPopuli, and MLS data to support 1,162 language and perform 300K updates; we denote this as multi-domain training. During fine-tuninig, the data of each language and dataset is balanced similar to pre-training (§4.2; using $\beta_D = 0$ and $\beta_L = 0.3$). The validation set for this model is the concatenation of the dev sets of each dataset for every language.

We evaluate this single model on FLEURS, CommonVoice, VoxPopuli and MLS. For FLEURS we measure average CER on the 102 languages of FLEURS, for CommonVoice WER over 76 languages, for VoxPopuli WER over 14 languages and for MLS WER over eight languages. We also fine-tune MMS (1B) on each benchmark individually (single-domain training).[21] During inference, we use n-gram models trained on CommonCrawl data.

Table 5 shows that the multi-domain model (MMS-lab+FL+CV+VP+MLS) can perform very competitively in several settings: For both FLEURS and CommonVoice it outperforms prior work as well the single-domain baselines and for VoxPopuli and MLS it is slightly worse than the single-domain baselines. For VoxPopuli, the MMS model it is outperformed by Maestro which supports a much smaller number of languages and on MLS other approaches are better which attribute to a focus on fewer languages. Whisper results are not strictly comparable due to the different normalization but it appears to perform better on MLS due to a focus on head languages. Overall, this demonstrates that a combined model supporting well over 1,000 languages can perform in aggregate competitively on a range of benchmarks.

	#lang	FLEURS	CV	VP	MLS
Prior Work					
VoxPopuli (Wang et al., 2021)	1			15.3	
Maestro (Chen et al., 2022b)	14			8.1	
RNN-T 1B (Li et al., 2021)	15				7.9
Whisper (Radford et al., 2022)	99			*13.6	*7.3
ML-IO (Tjandra et al., 2022b)	70				7.5
USM-M (Zhang et al., 2023a)	102	6.5			
This Work - Single-Domain training					
FL	102	6.4			
CV	76		19.7		
VP	14			10.3	
MLS	8				8.7
This Work - Multi-Domain training					
MMS-lab+FL+CV+VP+MLS	1,162	6.2	19.6	10.6	9.0

Table 5: **Evaluation of MMS on Multilingual Benchmarks.** MMS is fine-tuned on MMS-lab, FLEURS, CommonVoice, Voxpopuli and MLS. We report CER on FLEURS and WER on the other benchmarks. Results are averaged over all languages of a benchmark and (*) indicates results with a different data normalization which does not enable a strict comparison to other results (Radford et al., 2022). Our models use n-gram language models trained on Common Crawl during inference. For each approach, we show the number of languages an individual models supports and some prior results are based on multiple models, e.g., Wang et al. (2021).

5.5 Evaluation on 1,107 Languages

Finally, we evaluate the multi-domain model trained on MMS-lab, FLEURS, CommonVoice, VoxPopuli and MLS on the test sets of all 1,107 languages in MMS-lab. We measure character error rate and group languages into six geographical regions covered by MMS-lab: Asia, North America, South America, Europe, Africa and the Pacific region. In order to get a broader sense of quality, we measure the number of languages for which CER ≤ 5. This indicates for how many languages the model makes on average no more than one error in twenty characters. While this measure is very coarse, it enables us to get a sense of quality across such a large number of languages.

Table 6 shows that our model meets the CER quality threshold for 96% of the 1,107 languages. The region with the lowest rate is Africa at 91% and we attribute this in part to different writing scripts. We note that this metric is by far not perfect as it imposes the same threshold for every language which may not be appropriate due to different character

[20] RNN-T may benefit particularly from pre-training on unlabeled text data in the case of USM-M, effectively training a strong neural language model.

[21] Models trained on FLEURS data were trained for 300K updates, for MLS models we found 50k updates to work well, CommonVoice and VoxPopuli models were trained for 200K updates.

	#lang	CER	CER ≤ 5	%
Asia	335	1.6 ± 0.1	330	99%
South America	136	1.5 ± 0.2	132	97%
North America	144	2.2 ± 0.2	139	97%
Europe	41	1.7 ± 0.4	40	98%
Africa	363	2.9 ± 0.2	331	91%
Pacific	88	1.7 ± 0.5	87	99%
	1,107	2.1 ± 0.1	1,059	96%

Table 6: **ASR Evaluation on 1,107 Languages.** We evaluate the MMS multi-domain model trained on MMS-lab, FLEURS, CommonVoice, VoxPopuli, and MLS supporting 1,162 languages (§5.4) Results are in terms of the average character error rate on the MMS-lab test sets for the languages of different geographical regions. We also show the number of languages for which the model achieves CER less than five, indicating systems which on average produce no more than one incorrect character every twenty characters. All results are shown with confidence interval 95%.

sets etc. Also, many of the recordings in MMS-lab are single speaker which means that both the training data and the test data contains utterances with the same voice for a particular language. While this makes evaluation challenging, we hope that this analysis gives a sense that the model can be used to transcribe a wide variety of languages.

6. Language Identification

Language Identification (LID) is the task of determining the language which is spoken in a given utterance. This has several important applications: despite much work on multilingual speech recognition (Burget et al., 2010; Lin et al., 2009; Heigold et al., 2013; Bourlard et al., 2011; Cho et al., 2018; Toshniwal et al., 2018; Kannan et al., 2019; Li et al., 2019; Pratap et al., 2020b), many deployed systems are still trained on data in a single language despite the need to transcribe speech in different languages. It is therefore crucial to route utterances to the correct system and this routing depends on an LID model. Moreover, much work on mining speech data from the web relies on LID, including some of the most recent large-scale weakly-supervised work (Radford et al., 2022). Training a language identification system requires speech data for which the spoken language is known, however, the most diverse public corpora span no more than about 100 languages (Valk and Alumäe, 2020; Conneau et al., 2022).

In this section, we first describe our methodology to build LID models (§6.1), then evaluate the feasibility of training LID systems using MMS-lab-U and MMS-unlab data compared data from existing corpora (§6.2) and then build LID models with 40x more languages compared to existing systems (§6.3).[22]

[22] Note that we use MMS-lab-U instead of MMS-lab because it supports more languages and we do not require the data to be paired with transcriptions for LID (§3.1.1).

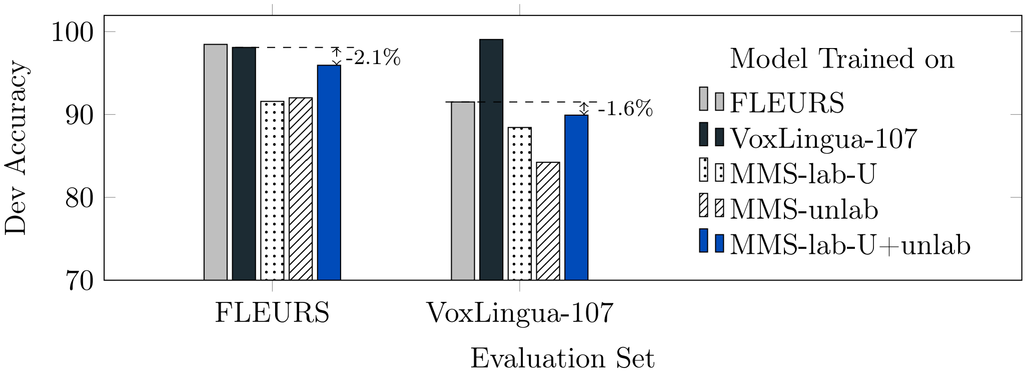


Figure 13: **LID Performance with Different Training Datasets.** Models trained on MMS-lab-U+unlab are very competitive to models trained on existing data (FLEURS, VoxLingua-107) when evaluated out-of-domain (comparison in dashed line). We train models on data from MMS-lab-U, MMS-unlab, FLEURS and VoxLingua-107 on a common subset of 72 languages and evaluate on FLEURS and VoxLingua-107.

6.1 Training Setup

We train models by fine-tuning the MMS (1B) pre-trained model (§4.2) for language identification. This is done by stacking a linear classifier on top of the pre-trained model, which maps to the set of possible languages for a particular task, followed by fine-tuning all parameters, including the pre-trained model.

We optimize models with Adam with exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.98$ and a tri-state learning rate schedule where the learning rate is warmed up for the first 10% of updates, held constant for the next 40% and then linearly decayed for the remainder of training (Kingma and Ba, 2015). During development, we experiment with different hyper-parameters and perform final model selection based on development set accuracy. We experiment with different learning rates (1×10^{-5}, 3×10^{-5}, 3×10^{-6}, 5×10^{-6}, 7×10^{-6}), training updates (10K, 20K, 30K, 40K, 50K) and batch sizes (1.5min, 3min, 6min). We train models on 16 GPUs.

To balance the different languages and corpora during training, we first balance the data of every language in the different corpora, using sampling parameter β_L. This is followed by balancing each language using the resampled data of the first step, using sampling parameter β_D with the sampling distribution outlined in §4.1. We experiment with β_L and β_D settings 0, 0.3, 0.5, 0.7 and 1.

When we train models on multiple datasets, then we use a development set containing up to 30 minutes of data for each language and we sample an equal amount of data from each corpus. For models supporting 1K, 2K or 4K languages, we reduce the amount of data per language to 15 min, 7min and 3min to enable faster development.

6.2 Comparison to Existing Datasets

We first assess the effectiveness of training LID models on MMS-lab-U and MMS-unlab data compared to existing LID training data (FLEURS, VoxLingua-107). Performance is evaluated both on the FLEURS and VoxLingua-107 benchmarks. In this setting, FLEURS and VoxLingua-107 models are at an advantage because there is no domain shift compared to systems trained on MMS-lab-U and MMS-unlab data.

We are particularly interested in the setting where models trained on existing corpora are evaluated out-of-domain, i.e., the performance of a model trained on VoxLingua-107 evaluated on FLEURS or a model trained on FLEURS evaluated on VoxLingua-107 data, and how this compares to models trained on MMS-lab-U/MMS-unlab. To enable a controlled comparison, we report LID accuracy on the 72 languages supported by all considered datasets and train only on data of these languages.

The results (Figure 13) show that MMS-lab-U+unlab enables LID with slightly lower performance compared to systems trained on existing datasets when both are evaluated out-of-domain: on FLEURS evaluation data the gap is 2.1% compared to a VoxLingua-107 model and on VoxLingua-107 evaluation data MMS-lab-U+unlab trails a FLEURS model by 1.6%. Combining MMS-lab-U and MMS-unlab works particularly well as MMS-unlab is more varied which improves performance. Naturally, models trained on in-domain training data (FLEURS or VoxLingua-107) perform best. We conclude that MMS-lab-U and MMS-unlab enable good quality LID models while enabling LID for many more languages as we will demonstrate next.

6.3 Scaling Language Identification to 4,017 Languages

Next, we scale spoken language identification from about 100 languages to 4,017 languages by combining MMS-lab-U, MMS-unlab, FLEURS, and VoxLingua-107 data. Our primary goal is to understand how accuracy is impacted as models support more and more languages. We start with 126 languages, the union of the languages in both FLEURS and VoxLingua-107, and then add languages from MMS-lab-U+unlab, roughly doubling the number of languages at every experiment. Languages are sorted by descending speaker count and we add the most spoken languages first.[23]

Performance is evaluated on FLEURS and VoxLingua-107 which is in-domain with respect to the models we train. To get a sense of how the models perform on domains not seen in the training data, we also evaluate on two other datasets which are out-of-domain with respect to the training data domain: first, BABEL is conversational telephone speech data and we evaluate on 23 African and Asian languages (Gales et al., 2014).[24] Second, VoxPopuli (Wang et al., 2021) consists of parliamentary speech in 25 languages from the European parliament. We evaluate on 2.5 hours of data for each language sampled from the VoxPopuli unlabeled data portion.

Table 7 shows that MMS models scale very well: increasing the number of languages from 126 to 4,017 results in a modest performance drop of just 0.3% on FLEURS and no drop on VoxLingua-107. Out-of-domain we observe a drop of 3.6% on BABEL and 0.2% on VoxPopuli. The results are also competitive to models trained only on in-domain data: On FLEURS evaluation data, the 4,017 language MMS model performs 1% better than the baseline trained only on FLEURS data and is only 0.2% behind the baseline trained on both

[23] We obtain the number of speakers of each language from `https://www.ethnologue.com`

[24] Amharic, Assamese, Bengali, Cantonese, Cebuano, Georgian, Guarani, Haitian Creole, Igbo, Javanese, Kazakh, Lao, Lithuanian, Dholuo, Mongolian, Pashto, Swahili, Tagalog, Tamil, Telugu, Turkish, Vietnamese, Zulu. We evaluate on test utterances longer than 10 seconds.

	#lang	in-domain		out-of-domain	
		FLEURS (102 lang.)	VL (33 lang.)	BABEL (23 lang.)	VoxPopuli (25 lang.)
Prior Work					
mSLAM (Bapna et al., 2022b)	102	77.7	-	-	-
Whisper (Radford et al., 2022)	82	64.5	-	-	-
ASRL (Chen et al., 2023)	102	95.9	-	-	-
XLS-R (Babu et al., 2022)	107	-	94.3	-	-
SpeechBrain (Ravanelli et al., 2021)	107	-	93.3	-	-
AmberNet (Jia et al., 2022)	107	-	95.3	-	-
Our Baselines (based on existing datasets)					
MMS (FL)	102	96.2	-	-	-
MMS (VL)	107	-	94.7	-	-
MMS (FL + VL)	126	97.4	94.3	78	87.8
This Work					
MMS (MMS-lab-U+unlab+FL+VL)	126	97.5	93.9	84.1	87.3
	256	97.2	93.4	80.1	87.6
	512	96.8	92.9	81.6	85.6
	1,024	97	92.8	80.5	86.2
	2,048	97.3	92.8	81.5	86.6
	4,017	97.2	93.9	80.5	87.1

Table 7: **Scaling LID to 4,017 Languages.** We show test accuracy of LID models trained on an increasing number of languages using data from FLEURS (FL), VoxLingua-107 (VL) and MMS-lab-U+unlab; smaller language subsets are included in the larger subsets. There is little performance degradation when scaling to more languages. We also show results from the literature as well as baselines trained only on FL or VL.

FLEURS and VoxLingua-107. On VoxLingua-107 evaluation data, the gap is 0.4-0.8%. This shows that scaling LID to 4,017 can result in models with competitive performance to models trained on much fewer languages.

7. Speech Synthesis

As a final downstream task we consider speech synthesis or text-to-speech (TTS) where models output speech for a corresponding input text. Most prior work focuses on English using clean corpora and high quality phonemizers (Tan et al., 2021). These resources are only available for a small number of languages which is why expanding TTS to 1,107 languages requires different choices. Black (2019) built TTS systems for 699 languages using data from a similar source as MMS-lab. Encouraged by our initial results indicating higher quality data (§3.3.1), we extend their work by building models for 1,107 languages.

We first describe the model architecture on which we build (§7.1), how we pre-process the data to train TTS models (§7.2) and how we evaluate our models (§7.3). Next, we present an ablation of our design choices compared to a highly optimized setup used for English

(§7.4). We also measure performance when synthesizing out-of-domain data for 61 languages of the FLEURS benchmark (§7.5), and finally present results for all 1,107 languages (§7.6).

7.1 Text-To-Speech Model

Our Text-to-Speech (TTS) model is based on VITS (Kim et al., 2021), which is one of the state-of-the-art TTS approaches. While VITS has previously been applied in a multilingual setting for English, Portuguese, and French (Casanova et al., 2022), we scale it to 1,107 languages.

VITS is an end-to-end speech generation network that predicts the raw speech waveform from a text sequence. It can be viewed as a conditional variational auto-encoder (VAE; Kingma and Welling 2013) that estimates audio features based on the input text. The spectrogram-based acoustic features are generated by a flow-based sub-network, which includes multiple affine coupling layers (Dinh et al., 2017) and a text encoder. The waveform is decoded using a stack of transposed convolutional layers that have been adapted from HiFi-GAN (Kong et al., 2020). The model is trained end-to-end with a combination of losses derived from variational lower bound and adversarial training. During inference, the text encodings are upsampled based on an internal duration prediction module and then mapped into the waveform using a cascade of the flow module and HiFi-GAN decoder.

We train separate VITS models for each language. Most of our hyperparameters are identical to the VITS model trained on LJSpeech (Ito and Johnson, 2017; Kim et al., 2021) except for these differences: Instead of training the model for 800K steps, we train each model for 100K steps using eight V100-GPUs with a batch size of 64 per GPU. This setup was only slightly worse than the original configuration but reduced training time by approximately eight times which made training a large number of TTS systems feasible (§7.3). We experimented with different learning rate settings and found that the original learning rate schedule of VITS worked best.

7.2 Text and Speech Data Pre-processing

Data Selection. To train models we use the MMS-lab dataset which provides paired speech and text data. For most languages we have a single recording of the New Testament, however, for 99 languages multiple recordings are available (§3.1.1). For these languages, we choose a single recording in order to avoid introducing additional speakers into the training data. To choose a recording, we train ASR models on the data and select the recording based on which the corresponding ASR model achieves the lowest CER on a held-out set of an out-of-domain evaluation set. If no out-of-domain evaluation set is available, we choose a random recording. Finally, if there are both drama and non-drama recordings, then we consider only non-drama recordings.

Text Representations. Most current TTS models convert text to phonemes using grapheme-to-phoneme tools such as g2p (Park and Kim, 2019) which rely on a lexicon to map input characters to phonetic representations. However, such lexicons are not available for most low resource languages since they require manual annotation. In order to scale TTS to over one thousand languages, we represent the input text as individual letters for languages with a small vocabulary. For languages with 200 or more characters, we use a uroman encoding (Hermjakob et al. 2018; §3.1.4). We validated this strategy on the languages of FLEURS where we found that letter-based models outperform uroman-based systems across all languages, except for Amharic and Korean, which both have large character sets of between 200-1,000 characters. We therefore used this mixed strategy.[25]

Speech Data Pre-processing. For drama recordings, we remove background music to enhance the quality of TTS models. We use a denoiser model (Defossez et al., 2020) to remove background music. We also noticed that some utterances contain multiple speakers, usually voicing different characters in the read stories. We use a simple heuristic to detect those utterances and remove them from the training data. The heuristic computes the variance of the pitch on voiced frames and removes utterances with high variance in the pitch; the pitch estimation is based on Mauch and Dixon (2014). We found that removing 15% of the utterances with the highest pitch variance in each recording removed many multi-speaker utterances.

7.3 Evaluation Methodology

Evaluation of speech synthesis is not straightforward even in the most researched English setting comprising a clean corpus with single speaker data (Ito and Johnson, 2017) and expanding evaluation to a large number of languages poses additional challenges. Similar to prior work, we rely on both automatic metrics (MCD and ASR) and human studies which we detail next.

Mel-Cepstral Distortion (MCD). Mel-cepstral distortion is an automatic metric which measures the closeness of synthesized speech to a human utterance for the same text in terms of the warping distance of mel frequency cepstral coefficients. There are two disadvantages of MCD: first, it is only meaningful when the voice and prosody of the speaker as well as recording conditions in the training data matches the evaluation data because the mel cepstral sequences are sensitive to these aspects. This prevents evaluation on data outside the MMS-lab domain. Second, it is not well suited to measuring the intelligibility of the TTS output. We address the former by focusing MCD evaluation on MMS-lab data and the latter via the next metric.

Automatic speech recognition (ASR). Transcribing the synthesized speech with automatic speech recognition and then measuring the error rate has the potential to address both issues of MCD: it can be measured on evaluation sets which are out-of-domain and it captures the content of the synthesized speech. Specifically, we synthesize both in-domain data from the MMS-lab evaluation sets as well as out-of-domain data from the FLEURS corpus. Next, we apply ASR models and compute CER of the ASR model output with respect to the input text of the TTS system. Low CER indicates that the TTS model is able to capture the content of the input text. Unless otherwise stated, we use ASR models trained on FLEURS data.

Mean Opinion Score (MOS). Finally, we evaluate models using MOS and because it is very hard to find human raters proficient for a large number of languages, we ask raters to

[25] For all 1,107 languages, the following languages use a uroman encoding: Amharic, Gumuz, Korean, Sebat Bet Gurage and Tigrinya.

train	train	text	ASR (CER)			MOS		
upd	data	repr.	MMS-lab	LJS	FLEURS	MMS-lab	LJS	FLEURS
Natural speech			4.4	4.3	9.3	3.89 ± 0.06	3.96 ± 0.06	3.37 ± 0.07
800K	LJS	phon.	5.5	4.9	5.9	3.87 ± 0.08	3.82 ± 0.06	3.73 ± 0.07
100K	LJS	phon.	6.3	4.9	6.3	3.64 ± 0.09	3.74 ± 0.07	3.66 ± 0.07
100K	MMS-lab	phon.	7.2	6.8	7.9	3.68 ± 0.07	3.51 ± 0.08	3.54 ± 0.07
100K	MMS-lab	chars	7.2	9.2	10.0	3.58 ± 0.08	3.45 ± 0.08	3.34 ± 0.09

Table 8: **Ablation of Training Setup.** Our design choices (last row) lead to a moderate quality reduction compared to the standard VITS setup (second row) but enable scaling TTS to 1,107 languages. The standard VITS setting performs 800K training updates on clean LJSpeech data using a high quality phonemizer while as MMS is trained on MMS-lab data for fewer updates and with characters. Natural speech results are based on the human utterances of each dataset. We report ASR CER and MOS scores from a human study with confidence interval 95% on the English development sets of MMS-lab, LJSpeech and FLEURS.

judge the fidelity of the synthesized samples together with how natural the speech sounds. We rely on ASR to get a sense of how much of the content is preserved in the TTS outputs and consider the MOS score for generation quality and naturalness. We evaluate 50 samples per method of each language, collect ten judgements per sample and ask raters to judge the quality from 1 (lowest quality) to 5 (highest quality). Results are reported in terms of the mean score as well as the 95% confidence interval. We use the CrowdMOS (Ribeiro et al., 2011) package with the recommended recipes for detecting and discarding inaccurate ratings. We do not require raters to be able to speak the respective language (except for English) as it is very hard to find raters that speak all the languages we consider. We rely on the ASR error rate to get a sense of content preservation in the TTS models.

7.4 Evaluation of Design Choices

7.4.1 Training Setup

We first analyze the quality of our training setup (§7.1) and compare it to the common VITS setup (Kim et al., 2021) which trains models for up to 800K updates on the clean LJSpeech corpus (Ito and Johnson, 2017) with text data pre-processed by a high-quality phonemizer (Park and Kim, 2019). This contrasts to our setup where we train the same model for 100K updates on the MMS-lab data using letters.

We evaluate performance on the development sets of MMS-lab which is in-domain for MMS-lab models, LJSpeech (LJS; Ito and Johnson 2017) which is in-domain for the original VITS setup and FLEURS which is out-of-domain for both.[26] The audio of FLEURS frequently contains high levels of noise and reverberation and it is relatively uncommon to use it for TTS, however, it does cover a large number of languages and it is out-of-domain with respect

[26] For measuring CER on LJSpeech, we remove punctuation and capitalization of both references and hypothesis using the normalization of Radford et al. (2022).

	ASR (CER)			MOS		
	MMS-lab	LJS	FLEURS	MMS-lab	LJS	FLEURS
Natural speech	4.4	4.3	9.3	3.89 ± 0.06	3.96 ± 0.06	3.37 ± 0.07
no background music	7.2	9.2	10.0	3.51 ± 0.07	3.52 ± 0.08	3.41 ± 0.09
background music	11.8	15.6	16.1	3.24 ± 0.08	2.98 ± 0.08	3.01 ± 0.08
+ denoise	10.8	13.2	14.4	3.32 ± 0.08	3.18 ± 0.07	3.16 ± 0.08
+ denoise + filter	7.8	10.8	11.9	3.47 ± 0.07	3.32 ± 0.08	3.12 ± 0.07

Table 9: **Ablation of Building TTS Models using Data with Background Music.** Curating the training data containing background music results in TTS systems which approach the performance of models trained on recordings without background music. We show MOS ratings as well as ASR CER on three different benchmarks for English data for the development sets of MMS-lab, LJSpeech and FLEURS. The TTS model labeled "no background music" is identical to the last row in Table 8 and we use the MMS-lab development set of the non-drama recording.

to the MMS-lab training data which makes it an interesting evaluation set, particularly for the experiment in §7.5.

Table 8 shows that our reduced setup (row 5) generally performs less well than the highly optimized setup of VITS for LJSpeech (row 2; Kim et al. 2021) and each design choice (fewer training updates, MMS-lab training data and character inputs) leads to a reduction in quality. In terms of character error rate, the degradation is most pronounced on out-of-domain settings with respect to our reduced setup (row 5) and less so on the in-domain MMS-lab development set. We stress that these choices enable scaling TTS to over 1,000 languages at manageable compute requirements and no need for language-specific text processing tools.

Note that the CER of almost all models on FLEURS data is lower than for the corresponding natural speech and that the MOS scores of the human reference utterances is also lower than for other datasets. We attribute this to the high levels of noise and reverberation in the FLEURS audio which results in increased CER for the original samples compared to the CER of the synthesized samples.

7.4.2 Data with Background Music

Next, we ablate how we build speech synthesis models based on recordings with background music which is a setting that applies to about 38% of the languages (§3.1.1). For this purpose, we train a model on an English recording with background music before and after the pre-processing steps outlined in §7.2 and compare this to a model trained on another English recording that does not contain any background music to start with (no background music). The pre-processing denoises the data and removes utterances with multiple speakers.

The results (Table 9) show that both denoising and removing samples with multi-speaker utterances results in performance improvements compared to the original data with background music. Both steps reduce the CER gap to models trained on no background music data by 69-87% relative to the CER of the system trained on data with background

	ASR (CER)		MOS	
	TTS	ref	TTS	ref
In-domain	11.1	9.2	3.51 ± 0.11	3.61 ± 0.11
Out-of-domain	11.3	8.8	3.52 ± 0.11	*3.33 ± 0.12

Table 10: **TTS in-domain and Out-of-domain Evaluation on 61 Languages.** We synthesize the test sets of MMS-lab (in-domain) and FLEURS (out-of-domain) to measure character error rate (CER) and collect human judgements (MOS) for both the model outputs (TTS) and human utterances (ref). The MMS models are robust with little performance degradation when applied to general domain data (out-of-domain): they retain much of the original content (low difference between CER of TTS and ref) and the systems produce outputs with good prosody and sound quality (low difference between MOS scores between TTS and ref). We show MOS scores with confidence interval 95%. (*) human judges assigned low ratings since the human reference audio in FLEURS contains a lot of variability compared to synthesized speech or the MMS-lab reference speech (in-domain ref).

music. MOS scores also increase after pre-processing, sometimes close to the level of the data with no background music on MMS-lab evaluation data (3.47 vs. 3.51).

7.5 Out-of-Domain Evaluation

The MMS-lab data is from a particular narrow domain (§3.1.1) which poses the question of whether TTS models trained on this data will generalize to other domains. To get a better sense of this, we train speech synthesis models and evaluate their quality on both in-domain and out-of-domain data. As in-domain data we use the test sets of MMS-lab and as out-of-domain data we use FLEURS. This enables evaluation on 61 languages of the FLEURS benchmark (FLEURS-61) which are covered by MMS-lab data. However, a downside of FLEURS is that it contains high levels of noise which makes it challenging when comparing synthesized audio to the human reference audio.

Table 10 shows that MMS models are robust to domain shift: the CER of synthesized speech (TTS) is only slightly higher out-of-domain compared to in-domain and MOS scores for the synthesized samples are nearly identical. We note that the in-domain and out-of-domain settings are measured on different data which does not enable strong claims about identical performance. The systems also retain much of the original content as the small difference in CER between TTS and human utterances.

The MOS scores also indicate that our systems have lower sound quality compared to human utterances but the difference is not very large on in-domain data (3.51 vs. 3.61). Unfortunately, out-of-domain MOS scores for the references are affected by the noisy speech in the FLEURS audio as noted earlier. We conclude that TTS models trained on MMS-lab data perform well out-of-domain.

7.6 Evaluation on 1,107 Languages

Finally, we train models for all languages of MMS-lab and focus on in-domain evaluation since finding out-of-domain evaluation data for such a large number of languages is difficult.

	#lang	MCD	ASR (CER)		TTS CER	%
			TTS	ref	≤ 5	
Asia	335	4.30 ± 0.1	3.1 ± 0.2	1.9 ± 0.1	296	88%
South America	136	4.10 ± 0.1	2.6 ± 0.2	1.8 ± 0.1	129	95%
North America	144	4.12 ± 0.1	3.8 ± 0.8	2.4 ± 0.2	125	87%
Europe	41	4.33 ± 0.2	3.0 ± 0.3	1.9 ± 0.2	39	95%
Africa	363	4.34 ± 0.1	4.1 ± 0.2	2.6 ± 0.1	277	76%
Pacific	88	4.72 ± 0.2	3.4 ± 1.3	1.8 ± 0.2	79	90%
	1,107	4.30 ± 0.0	3.5 ± 0.2	2.2 ± 0.1	945	85%

Table 11: **TTS Evaluation on 1,107 Languages.** The majority of MMS TTS models can synthesize speech which preserves most of the content as per the ASR character error rates (TTS CER ≤ 5). We report MCD and character error rate for the synthesized outputs (TTS) of the MMS-lab test sets as well as the human references (ref). We also show the number of systems which achieve CER less than five, indicating systems which on average produce no more than one incorrect character every twenty characters. Results are shown with confidence interval 95%.

Specifically, we measure MCD and ASR CER on the MMS-lab test sets. To be able to evaluate ASR quality on all languages, we use ASR models trained on MMS-lab data which results in much lower error rates. Similar to §5.5, we group results into six geographical regions covered by MMS-lab: Asia, North America, South America, Europe, Africa and the Pacific region.

To get an overall sense of how many models are of good quality, we measure whether the model for a particular language has ASR CER ≤ 5. This indicates the number of systems which make on average no more than one error in twenty characters. While this measure is by far not perfect, it enables us to get a broad sense of quality across a large number of languages.

Table 11 shows that about 85% of the 1,107 languages meet the CER quality threshold. South American and European languages achieve the highest rate at 95% and African languages the lowest rate of 76%. This is in part driven by different writing scripts. The ASR character error rates are generally low because the error rates are based on ASR models trained on MMS-lab data.

8. Bias Analysis and Ethical Considerations

Training machine learning models on religious texts may introduce biases and requires ethical considerations. In this section, we analyze whether our models are biased to perform better for different genders (§8.1), if the language produced by our models is religiously biased (§8.2 and finally, we discuss ethical considerations of using religious texts in research (§8.3).

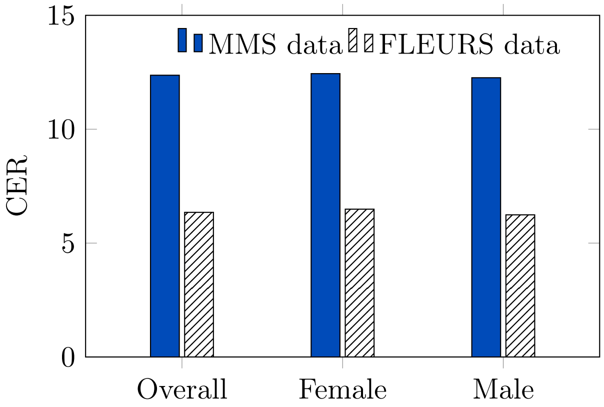


Figure 14: **Analysis of Gender Bias.** We compare the character error rate (CER) of automatic speech recognition models trained on MMS-lab data and FLEURS data for male and female spakers. Results are on the development sets of 27 languages of the FLEURS benchmark for which MMS-lab provides data and for which there are at least 50 samples for each gender.

8.1 Understanding Gender Bias

Most speakers in MMS-lab dataset appear to be male and this bears the risk of machine learning models trained on this data performing better for male speakers. To understand whether the models trained on our datasets (§3) exhibit gender bias, we perform the following experiment: we evaluate models on the development set of FLEURS which contains metadata indicating the gender of the speaker and we use this information to report performance for each gender. Using this split, we evaluate the accuracy of ASR models trained on MMS-lab restricted to languages of FLEURS for which MMS-lab provides data (61 languages) and for which there are at least 50 samples for each gender (27 of these 61 languages).

Figure 14 shows that the average character error rate over these 27 languages is very similar, both for the MMS model and the model trained on FLEURS data. There can be significant differences between genders within a particular language, but both models appear to be equally affected (see Table A4). On a per language basis, male speakers have a higher error rate for 14 languages while as female speakers have a higher error rate for the remaining 13 languages. We conclude that our models exhibit similar gender bias to models trained on FLEURS data which is general domain data.

8.2 Understanding Language Bias

The datasets created as part of this study are from a particular narrow domain and machine learning models estimated on this data may exhibit certain biases. In this section, we examine the extent to which automatic speech recognition models trained on MMS-lab data output biased language.

Methodology. The general methodology of our analysis is to identify a set of biased words in each language and to measure the rate at which biased words are produced by ASR models. We compare models trained on MMS-lab data and models trained on FLEURS, a corpus of

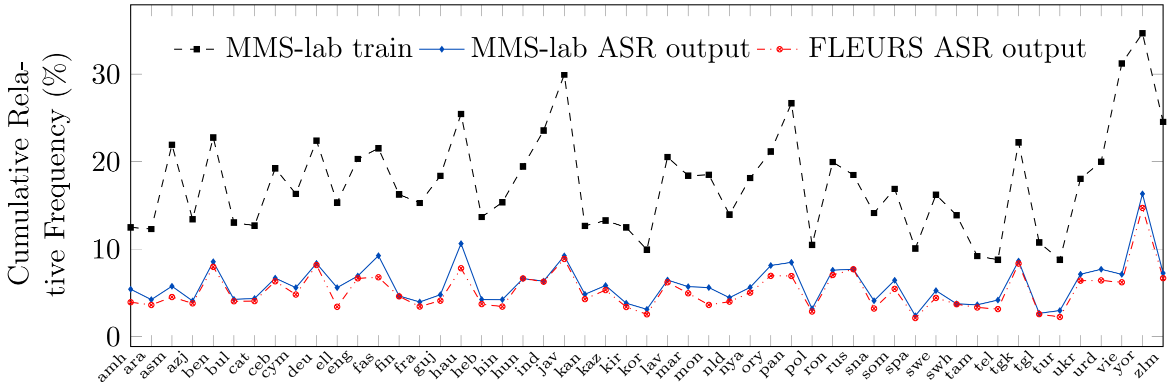


Figure 15: **Analysis of Language Produced by MMS ASR Models.** MMS models generate biased words at a slightly increased rate of 0.7% absolute compared to models trained on FLEURS data. We focus on words that occur at least twice as often in the MMS-lab training data compared to Common Crawl (in terms of relative frequency). We compare the cumulative relative frequency of these words in the MMS model output and the output of ASR models trained on FLEURS data. ASR outputs are based on transcribing the development sets of 51 languages.

people reading Wikipedia articles in different languages. Both sets of models are tasked to transcribe the FLEURS development set and we are interested in whether models trained on MMS-lab data are more likely to use biased language compared to models trained on FLEURS data.

Identifying Biased Words. We were not able to find speakers for most of the considered languages of this study and therefore use the following automatic procedure to determine religious words: for each word that occurs in the training data of MMS-lab, we compare the relative token frequency, that is, the rate at which the word type occurs in the MMS-lab data, to the relative token frequency in a general domain corpus; we use Common Crawl (Conneau et al., 2020b) as a general domain corpus. If the relative word frequency is at least twice as high in MMS-lab compared to Common Crawl, then we add it to the subset of words we include in our study. This enables us to evaluate on 51 languages of the FLEURS corpus since not all languages are covered by MMS-lab and we also need to find data in Common Crawl for each language. The automatic procedure has the added benefit of avoiding any potential biases introduced by human annotators.

Results. Figure 15 shows that the rate of biased words is much lower in the outputs of MMS models compared to the training data (MMS-lab ASR output vs. MMS-lab train). For many languages, MMS models generate these words at the same rate as the FLEURS models. On average the rate of biased words is 0.7% absolute higher for MMS compared to the FLEURS model. We interpret this as a slight bias as the difference between MMS-lab ASR output and FLEURS ASR output in Figure 15 shows.

We consulted with native speakers for languages which showed the largest discrepancies: for Mongolian (mon), a native speaker verified that most of the biased words in question are actually general language with no particular bias. For Persian (fas), only two out of the words the procedure identified were of religious nature and both were indeed over predicted: the Persian words for Jesus and spirit/ghost were predicted in four instances (on the entire development set) while the FLEURS model does not predict these words, however, the MMS model also predicts the word for hand 15 times more often than the baseline FLEURS model.

In English, the procedure identifies words such as *you*, *that*, *they* but it also includes *jesus*, which both models predict at the same rate. There is also *christ*, and *lord*, both of which are not predicted at all, or *men* which is predicted six times by the MMS model compared to five times by the FLEURS model. Our method of identifying biased words has low precision but it does capture words which are likely more used in religious contexts than otherwise.

8.3 Ethical Considerations and use of Religious Texts in Research

Our consultations with Christian ethicists concluded that most Christians would not regard the New Testament, and translations thereof, as too sacred to be used in machine learning. The same is not true for all religious texts: for example, the Quran was originally not supposed to be translated. There is also the risk of religious training data biasing the models with respect to a particular world view, however, our analysis of the language generated by our models suggests that the language produced by the resulting speech recognition models exhibit only little bias compared to baseline models trained on other domains (§8.2).

This project follows a long line of research utilizing the New Testament to train and evaluate machine learning models. The most related project is the CMU Wilderness effort (Black, 2019) which created speech synthesis models for 699 languages using speech and text data from similar sources as our datasets. For machine translation, researchers used data from the Bible both for training and evaluation (Christodouloupoulos and Steedman, 2015; McCarthy et al., 2020; NLLB Team et al., 2022). For speech processing, researchers trained speech synthesis models for ten African languages based on readings of the bible (Meyer et al., 2022).

9. Conclusions and Open Problems

We presented the first study which scaled speech technology to over one thousand languages. This has been made possible by the rapid progress in self-supervised speech representation learning which in turn enabled more sample efficient learning from labeled data. We presented how we collected datasets, pretrained models and then built models for automatic speech recognition, language identification and speech synthesis. This scaled the number of supported languages for several major speech tasks by between 10-40x. Going forward, we see several avenues for future work:

Scaling to even more languages and dialects. Even though, we built speech systems supporting between 1,100-4,000 languages, there are currently over 7,000 languages being spoken around the world today. Moreover, there are many more dialects which are often not adequately represented in the training data, even for high-resource languages such as English. This can lead to undesirable biases in the performance of these models (Koenecke et al., 2020).

Multi-task models. Another avenue is to train single models for several downstream tasks such as speech recognition, language identification etc. which can then all be performed by a single model. There has been work on a moderate number of languages (Radford et al., 2022) but we hope that this approach can be scaled to many more languages and with a smaller focus on head languages.

Tackling more speech tasks. While this study covered three different speech tasks, there are many more tasks involving speech data, such as speech translation, both to text and to speech, or keyword spotting, intent classification etc. We hope that future work will expand these tasks to many more languages as well.

Acknowledgments

We would like to thank Chloé Bakalar and Hubert Etienne for providing feedback on responsible AI evaluations, as well as the people who featured in the demo video: Filmawit Gebreegziabher, Gilbert Buada, Shanti Jha, Masoud Tavazoei, Akinniyi Akinyemi, Mari Silvia Marin, Amlan Roy, Kumar Rajarshi, Sugat Nayak, Jason Carlo Carranceja, Amit Prakash, Hangameh Vahabzadeh, Mohsen Dalil, Swati Verma, Edwina Anandita, Faisal Putra. We would like to thank Moto Hira for help in preparing the forced alignment tutorial in TorchAudio and Ping Yu for analyzing the Chinese language datasets for MMS. We thank Sainbayar Sukhbaatar for his analysis of the model transcriptions.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *Proc. of LREC*, 2020.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv*, 2016.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. In *Proc. Interspeech 2022*, pages 2278–2282, 2022. doi: 10.21437/Interspeech.2022-143.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *Proc. of ICLR*, 2020a.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proc. of NeurIPS*, 2020b.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Unsupervised speech recognition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Mandeep Baines, Shruti Bhosale, Vittorio Caggiano, Naman Goyal, Siddharth Goyal, Myle Ott, Benjamin Lefaudeux, Vitaliy Liptchinsky, Mike Rabbat, Sam Sheiffer, Anjali Sridhar, and Min Xu. Fairscale: A general purpose modular pytorch library for high performance and large scale training. `https://github.com/facebookresearch/fairscale`, 2021.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, Theresa Breiner, Vera Axelrod, Jason Riesa, Yuan Cao, Mia Xu Chen, Klaus Macherey, Maxim Krikun, Pidong Wang, Alexander Gutkin, Apurva Shah, Yanping Huang, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. Building machine translation systems for the next thousand languages. *arXiv*, abs/2205.03983, 2022a.

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*, abs/2202.01374, 2022b.

Alan W Black. Cmu wilderness multilingual speech dataset. In *Proc. of ICASSP*, 2019.

Hervé Bourlard, John Dines, et al. Current trends in multilingual speech processing. *Sadhana*, 2011.

Lindell Bromham, Russell Dinnage, Hedvig Skirgard, Andrew Ritchie, Marcel Cardillo, Felicity Meakins, Simon Greenhill, and Xia Hua. Global predictors of language endangerment and the future of linguistic diversity. *Nature Ecology and Evolution*, 2021.

Lukáš Burget, Petr Schwarz, et al. Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models. In *Proc. of ICASSP*, 2010.

Xingyu Cai, Jiahong Yuan, Yuchen Bian, Guangxu Xun, Jiaji Huang, and Kenneth Church. W-CTC: a connectionist temporal classification loss with wild cards. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=0RqDp8FCW5Z`.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pages 2709–2720. PMLR, 2022.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell. *arXiv*, 2015.

William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv*, abs/2104.02133, 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.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv*, abs/1604.06174, 2016.

William Chen, Brian Yan, Jiatong Shi, Yifan Peng, Soumi Maiti, and Shinji Watanabe. Improving massively multilingual asr with auxiliary ctc objectives. In *Proc. of ICASSP*, 2023.

Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro Moreno, Ankur Bapna, and Heiga Zen. Maestro: Matched speech text representations through modality matching. *arXiv*, abs/2204.03409, 2022b.

Jaejin Cho, Murali Karthick Baskar, et al. Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling. In *Proc. of IEEE SLT*, 2018.

Christos Christodouloupoulos and Mark Steedman. A massively parallel corpus: the bible in 100 languages. In *Proc. of LREC*, 2015.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv*, abs/1609.03193, 2016.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Proc. of NeurIPS*, 2019.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. Unsupervised cross-lingual representation learning for speech recognition. *arXiv*, abs/2006.13979, 2020a.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proc. of ACL*, 2020b.

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. *arXiv*, 2022.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. In *Interspeech*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. of NAACL*, 2019.

Sander Dieleman, Aäron van den Oord, and Karen Simonyan. The challenge of realistic music generation: modelling raw audio at scale. *Proc of NIPS*, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*. OpenReview.net, 2017.

David Doukhan, Eliott Lechapt, Marc Evrard, and Jean Carrive. Ina's mirex 2018 music and speech detection system. In *Proc. of MIREX*, 2018.

Ewan Dunbar, Mathieu Bernard, Nicolas Hamilakis, Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Eugene Kharitonov, and Emmanuel Dupoux. The zero resource speech challenge 2021: Spoken language modelling. In *Proc. of Interspeech*, 2021.

Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu. Exploring wav2vec 2.0 on speaker verification and language identification. *arXiv*, 2021.

Mark J. F. Gales, Kate M. Knill, Anton Ragni, and Shakti P. Rath. Speech recognition and keyword spotting for low-resource languages: Babel project research at cued. In *Spoken Language Technologies for Under-Resourced Languages*, 2014.

Alex Graves, Santiago Fernández, and Faustino Gomez. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. of ICML*, 2006.

Mutian He, Jingzhou Yang, Lei He, and Frank K. Soong. Multilingual byte2speech models for scalable low-resource speech synthesis. *arXiv*, abs/arXiv:2103.03541, 2021.

Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean. Multilingual acoustic models using distributed deep neural networks. In *Proc. of ICASSP*, 2013.

Ulf Hermjakob, Jonathan May, and Kevin Knight. Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4003. URL `https://aclanthology.org/P18-4003`.

James L Hieronymus. Ascii phonetic symbols for the world's languages: Worldbet. *JIPA*, 23: 72, 1993.

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*, pages 2790–2799. PMLR, 2019.

Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, et al. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*, 2021a.

Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: How much can a bad teacher benefit ASR pre-training? In *Proc. of ICASSP*, 2021b.

Keith Ito and Linda Johnson. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *Proc. of ICLR*, 2016.

Tahir Javed, Sumanth Doddapaneni, Abhigyan Raman, Kaushal Santosh Bhogale, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. Towards building asr systems for the next billion users. In *Proc. of AAAI CAI*, 2022.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, January 2011.

Fei Jia, Nithin Rao Koluguri, Jagadeesh Balam, and Boris Ginsburg. Ambernet: A compact end-to-end model for spoken language identification. *arXiv*, abs/2210.15781, 2022.

Jacob D Kahn, Vineel Pratap, Tatiana Likhomanenko, Qiantong Xu, Awni Hannun, Jeff Cai, Paden Tomasello, Ann Lee, Edouard Grave, Gilad Avidov, et al. Flashlight: Enabling innovation in tools for machine learning. In *International Conference on Machine Learning*, pages 10557–10574. PMLR, 2022.

Herman Kamper, Aristotelis Anastassiou, and Karen Livescu. Semantic query-by-example speech search using visual grounding. In *Proc. of ICASSP*, 2019.

Anjuli Kannan, Arindrima Datta, et al. Large-scale multilingual speech recognition with a streaming end-to-end model. In *Proc. of Interspeech*, 2019.

Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *Proc. of ICML*, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 2020.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *NeurIPS*, 2020.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. Ctc-segmentation of large corpora for german end-to-end speech recognition. In Alexey Karpov and Rodmonga Potapova, editors, *Speech and Computer*, pages 267–278, Cham, 2020. Springer International Publishing.

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 *Proc. of EMNLP*, 2022.

M. Paul Lewis, Gary F. Simon, and Charles D. Fennig. Ethnologue: Languages of the world, nineteenth edition. Online version: `http://www.ethnologue.com`, 2016.

Bo Li, Yu Zhang, Tara Sainath, Yonghui Wu, and William Chan. Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes. In *Proc. of ICASSP*, 2019.

Bo Li, Ruoming Pang, Tara N. Sainath, Anmol Gulati, Yu Zhang, James Qin, Parisa Haghani, W. Ronny Huang, Min Ma, and Junwen Bai. Scaling end-to-end models for large-scale multilingual asr. In *Proc. of ASRU*, 2021.

Xinjian Li, Florian Metze, David R Mortensen, Alan W Black, and Shinji Watanabe. Asr2k: Speech recognition for around 2000 languages without audio. *arXiv preprint arXiv:2209.02842*, 2022.

Tatiana Likhomanenko, Gabriel Synnaeve, and Ronan Collobert. Who needs words? lexicon-free speech recognition. In *Proc. of Interspeech*, 2019.

Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Paden Tomasello, Jacob Kahn, Gilad Avidov, Ronan Collobert, and Gabriel Synnaeve. Rethinking evaluation in ASR: are our models robust enough? In *Proc. of Interspeech*, 2020.

Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee. A study on multilingual acoustic modeling for large vocabulary asr. In *Proc. of ICASSP*, 2009.

Alexander H. Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. Towards end-to-end unsupervised speech recognition. *arXiv*, 2022.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *arXiv*, abs/2001.08210, 2020.

Loren Lugosch, Tatiana Likhomanenko, Gabriel Synnaeve, and Ronan Collobert. Pseudo-labeling for massively multilingual speech recognition. In *Proc. of ICASSP*, 2022.

M-AILABS. The m-ailabs speech dataset, 2018. URL `https://www.caito.de/2019/01/03/the-m-ailabs-speech-dataset/`.

Matthias Mauch and Simon Dixon. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, 2014.

Mark Mazumder, Colby Banbury, Josh Meyer, Pete Warden, and Vijay Janapa Reddi. Few-shot keyword spotting in any language. In *Proc. of Interspeech*, 2021.

Arya McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. The johns hopkins university bible corpus: 1600+ tongues for typological exploration. In *Proc. of LREC*, 2020.

Josh Meyer, David Ifeoluwa Adelani, Edresson Casanova, Alp Öktem, Daniel Whitenack Julian Weber, Salomon Kabongo, Elizabeth Salesky, Iroro Orife, Colin Leong, Perez Ogayo, Chris Emezue, Jonathan Mukiibi, Salomey Osei, Apelete Agbolo, Victor Akinode, Bernard Opoku, Samuel Olanrewaju, Jesujoba Alabi, and Shamsuddeen Muhammad. Bibletts: a large, high-fidelity, multilingual, and uniquely african speech corpus. *arXiv*, abs/2207.03546, 2022.

Tomaš Nekvinda and Ondrej Dusek. One model, many languages: Meta-learning for multilingual text-to-speech. In *Interspeech*, 2020.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, et al. No language left behind: Scaling human-centered machine translation, 2022.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL System Demonstrations*, 2019.

Kyubyong Park and Jongseok Kim. g2pe. `https://github.com/Kyubyong/g2p`, 2019.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *Proc. of ASRU*, 2011.

V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert. Wav2letter++: A fast open-source speech recognition system. In *Proc. of ICASSP*, 2019.

Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters. In *Proc. of Interspeech 2020*, 2020a.

Vineel Pratap, Anuroop Sriram, et al. Massively multilingual asr: 50 languages, 1 model, 1 billion parameters. *arXiv*, abs/2007.03001, 2020b.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. In *Proc. of Interspeech*, 2020c.

Vineel Pratap, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. Star temporal classification: Sequence modeling with partially labeled data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=ldRyJb_cjXa`.

Ting Qian, Kristy Hollingshead, Su-youn Yoon, Kyoung-young Kim, and Richard Sproat. A python toolkit for universal transliteration. In *Proc. of LREC*, 2010.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv*, 2022.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. Zero-infinity: Breaking the GPU memory wall for extreme scale deep learning. *arXiv*, abs/2104.07857, 2021.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François

Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

Flávio Ribeiro, Dinei Florêncio, Cha Zhang, and Michael Seltzer. Crowdmos: An approach for crowdsourcing mean opinion score studies. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2416–2419. IEEE, 2011.

Takaaki Saeki, Soumi Maiti, Xinjian Li, Shinji Watanabe, Shinnosuke Takamichi, and Hiroshi Saruwatari. Learning to speak from text: Zero-shot multilingual text-to-speech with unsupervised text pretraining. *arXiv*, abs/2301.12596, 2023a.

Takaaki Saeki, Heiga Zen, Zhehuai Chen, Nobuyuki Morioka, Gary Wang, Yu Zhang, Ankur Bapna, Andrew Rosenberg, and Bhuvana Ramabhadran. Virtuoso: Massive multilingual speech-text joint semi-supervised learning for text-to-speech. In *Proc. of ICASSP*, 2023b.

Setffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In *Proc. of Interspeech*, 2019.

Jiatong Shi, Dan Berrebbi, William Chen, Ho-Lam Chung, En-Pei Hu, Wei Ping Huang, Xuankai Chang, Shang-Wen Li, Abdelrahman Mohamed, Hung yi Lee, and Shinji Watanabe. Ml-superb: Multilingual speech universal performance benchmark. In *Proc. of Interspeech*, 2023.

Marlene Staib, Tian Huey Teh, Alexandra Torresquintero, Devang S Ram Mohan, Lorenzo Foglianti, Raphael Lenain, and Jiameng Gao. Phonological features for 0-shot multilingual speech synthesis. In *Interspeech*, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Proc. of NIPS*, 2014.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

Andros Tjandra, Diptanu Gon Choudhury, Frank Zhang, Kritika Singh, Alexis Conneau, Alexei Baevski, Assaf Sela, Yatharth Saraf, and Michael Auli. Improved language identification through cross-lingual self-supervised learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6877–6881. IEEE, 2022a.

Andros Tjandra, Nayan Singhal, David Zhang, Ozlem Kalinli, Abdelrahman Mohamed, Duc Le, and Michael L. Seltzer. Massively multilingual asr on 70 languages: Tokenization, architecture, and generalization capabilities. *arXiv*, abs/2211.05756, 2022b.

Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. Multilingual speech recognition with a single end-to-end model. In *Proc. of ICASSP*, 2018.

Jörgen Valk and Tanel Alumäe. Voxlingua107: a dataset for spoken language recognition. In *Proc. of SLT*, 2020.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Proc. of NIPS*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NIPS*, 2017.

Changhan Wang, Morgane Riviere, 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 *Proc. of ACL*, 2021.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211, 2018. doi: 10.21437/Interspeech.2018-1456. URL `http://dx.doi.org/10.21437/Interspeech.2018-1456`.

John Wells. Computer-coding the ipa: a proposed extension of sampa, 1995. URL `https://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm`.

Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Anjali Chourdia, Artyom Astafurov, Caroline Chen, Ching-Feng Yeh, Christian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z. Yang, Jason Lian, Jay Mahadeokar, Jeff Hwang, Ji Chen, Peter Goldsborough, Prabhat Roy, Sean Narenthiran, Shinji Watanabe, Soumith Chintala, Vincent Quenneville-Bélair, and Yangyang Shi. Torchaudio: Building blocks for audio and speech processing. *arXiv preprint arXiv:2110.15018*, 2021.

Su-Youn Yoon, Kyoung-Young Kim, and Richard Sproat. Multilingual transliteration using feature based phonetic method. In *Proc. of ACL*, 2007.

Yu Zhang, Ron J. Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, RJ Skerry-Ryan, Ye Jia, Andrew Rosenberg, and Bhuvana Ramabhadran. Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning. In *Interspeech*, 2019.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv*, 2023a.

Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv*, abs/2303.03926, 2023b.

Appendices

Appendix A. Forced Alignment

Given an input audio sequence $X = (x_1, ..., x_N)$, where N is the number of frames, a CTC alignment model produces an output sequence $Y = (\mathbf{y^1}, ..., \mathbf{y^T})$ of length T, where $\mathbf{y^t}$ denotes the posterior probability distribution over an alphabet $\mathcal{A}$ and blank token $\langle b \rangle$. Let $\mathcal{B}$ denote the collapsing function of CTC which collapses all repeating symbols and which removes all blanks for a given sequence. An alignment path $\boldsymbol{\pi} = \pi_1, ..., \pi_T$ in CTC for a given target label of length M, $L = (l_1, ..., l_M)$ where $\pi_t \in \mathcal{A} \cup \{\langle b \rangle\}$ and $l_i \in \mathcal{A}$ should satisfy $\mathcal{B}(\boldsymbol{\pi}) = L$.

Among the all the alignment paths which can be collapsed to the given target label L, forced alignment computes the best alignment path $\hat{\boldsymbol{\pi}}$ which maximizes the probability under the posterior distribution given by the acoustic model.

$$\hat{\boldsymbol{\pi}} = \underset{\boldsymbol{\pi} \in \mathcal{B}^{-1}(L)}{\arg\max}\, P(\boldsymbol{\pi}|X) \tag{2}$$

CTC assumes every output is conditionally independent of the other outputs given the input. We have

$$P(\boldsymbol{\pi}|X) = \prod_{t=1}^{T} P(\boldsymbol{\pi_t}|t, X) = \prod_{t=1}^{T} y^t_{\pi_t} \tag{3}$$

Equation 2 can be computed efficiently using dynamic programming and backtracking. An efficient version of this algorithm is implemented in flashlight (Kahn et al., 2022) on CPU. We implement a parallel version on CUDA, incorporating memory optimizations that offload memory to CPU, allowing it to process long audio files efficiently as shown in Algorithm 1.

Relationship to Other Alignment Generation Approaches. For dealing with noisy transcripts, a commonly used alternative to forced alignment is as follows (Povey et al., 2011): first segment the audio into shorter segments that can be input to an acoustic model composed with a language model and generate a transcription for each segment.[27] Then align the generated transcription with the original text to determine the audio segment for each word. If the generated transcriptions cannot be well aligned, then these segments are discarded.

The advantage of this approach is that it enables alignment when the audio and the text do not correspond entirely to each other. However, the alignments are performed on a segment per segment basis, whereas our forced alignment process performs a global alignment taking the entire sequence into account.

Appendix B. *n*-gram Language Models

We train 5-gram language models on Common Crawl data using KenLM (Heafield, 2011) for each language in FLEURS. For languages that do not use spaces to separate words, we train 20-gram character-level language models. These languages are Mandarin Chinese (cmn), Cantonese Chinese (yue), Japanese (jpn), Thai (tha), Lao (lao), Burmese (mya) and Khmer (khm). The text is pre-processed following § 3.1.2 and we also remove emojis.[28].

[27] `https://github.com/mozilla/DSAlign`

[28] `https://stackoverflow.com/a/33417311`

Algorithm 1 Pseudo code of our CTC Forced Alignment algorithm on GPU

```
Input: Posterior probabilities $\mathbf{y}$, target label $L$
Output: Alignment path $\pi$
$S \leftarrow 2 \times |L| + 1$
Create GPU matrices $\alpha_{\text{odd}}$, $\alpha_{\text{even}}$ of size $S$ each to store the maximum log probability of aligning the target upto the given node
B $\leftarrow$ 100; # buffer size for data copy from GPU to CPU
Create CPU matrix $\beta$ of size $T \times S$ for saving the label indices from previous time step used to compute $\alpha_{\text{odd}}/\alpha_{\text{even}}$
Create GPU matrix $\beta_{\text{buffer}}$ of size $B \times S$, where B is the buffer size, to store the values of $\beta$ temporarily in a buffer before being copied to CPU.
for $t = 1, \ldots, T$ do
    for all $l = 1, \ldots, S$ do in parallel
        If $t$ is odd, update $\alpha_{\text{odd}}[l]$ based on $\alpha_{\text{even}}$, $\mathbf{y^t}$ using dynamic programming and vice-versa
        Store the index $l'$ from previous time step used to update $\alpha_{\text{odd}}/\alpha_{\text{even}}$ in $\beta_{\text{buffer}}$
    end for
    if t % B == 0 or t == T then
        Copy $\beta_{\text{buffer}}$ to CPU matrix $\beta$ asynchronously
    end if
end for
Backtrack and compute $\pi$ from $\alpha_{\text{odd}}$, $\alpha_{\text{even}}$ and $\beta$
```

For word-level models, we limit the training data to 40GB and select the 250K most-frequent words as the vocabulary. For character-level models, we limit the training data to 6GB. We provide example commands used for training the LMs below:

```
# word-level LM
> kenlm/build/bin/lmplz -prune 1 2 3 4 5 -o 5 -limit_vocab_file vocab.txt -S 90% -T /tmp/ < input.txt > output.arpa [29]
# character-level LM
> kenlm/build/bin/lmplz -prune 0 0 0 0 0 1 1 1 2 3 -o 20 trie -S 90% -T /tmp/ < input.txt > output.arpa
```

For n-gram models trained on the FLEURS training data transcriptions, we build 15-gram character level language models without any pruning on all languages in FLEURS. For the comparison with Whisper, we only use the Common Crawl language models.

We use the CTC beam-search decoder from the Flashlight (Kahn et al., 2022) library for decoding our models. For decoding with word-level LMs, we use the lexicon-based decoder of Collobert et al. (2016); Pratap et al. (2019) and for character-level LMs, we use the lexicon-free beam-search decoder of Likhomanenko et al. (2019). We tune the language model weight and word insertion penalty on the validation set to select the best hyperparameters for decoding the test set.

[29] We use `-prune 1 1 2 3 4` if data size < 5GB and additionally use `-discount_fallback` if data size < 1GB

Appendix C. Comparison to Whisper

Table A1 shows a breakdown of the results into individual languages and Table A2 shows results with and without CC LM n-gram language models.

	Whisper medium	Whisper large-v2	MMS L-61 noLM	MMS L-61 CC LM	MMS L-61 noLM LSAH	MMS L-61 CC LM LSAH	MMS L-1107 noLM	MMS L-1107 CC LM	MMS L-1107 noLM LSAH	MMS L-1107 CC LM LSAH
Amharic	229.3	140.3	48.7	30.7	52.4	32.5	52.9	30.1	53.3	31.1
Arabic	20.4	16.0	34.9	19.6	35.8	19.9	44.0	23.4	41.3	21.0
Assamese	102.3	106.2	29.5	18.8	28.4	18.6	37.6	21.2	30.5	19.2
Azerbaijani	33.1	23.4	40.7	21.3	38.3	19.8	45.0	21.2	40.1	19.1
Bengali	100.6	104.1	19.7	11.6	20.0	12.1	25.0	12.5	23.5	12.1
Bulgarian	21.4	14.6	23.4	13.1	23.9	13.3	27.9	12.9	25.5	13.5
Burmese	123.0	115.7	22.2	14.2	22.3	14.5	29.2	20.2	24.5	16.0
Catalan	9.6	7.3	18.1	11.0	18.1	11.0	25.9	11.5	20.1	10.8
Dutch	9.9	6.7	26.9	13.7	26.4	14.3	38.1	14.9	27.6	14.5
English	4.4	4.2	23.8	10.7	24.8	11.8	38.8	12.2	27.8	12.3
Filipino	19.1	13.8	19.3	11.9	19.4	12.2	26.2	13.5	20.2	12.4
Finnish	13.9	9.7	26.4	22.5	26.9	23.1	32.3	22.2	28.8	23.1
French	8.7	8.3	24.3	13.7	24.5	14.1	35.8	15.4	29.3	15.0
German	6.5	4.5	22.5	13.2	22.3	13.7	38.4	13.1	22.5	13.3
Greek	19.0	12.5	40.8	14.0	40.5	13.6	57.5	13.0	40.1	13.6
Gujarati	104.8	102.7	23.0	13.0	22.7	12.8	73.9	56.4	24.0	12.8
Hausa	106.6	88.9	35.9	26.7	36.3	27.3	40.4	26.7	38.3	26.4
Hebrew	33.1	27.1	68.5	44.8	66.6	41.5	78.7	50.9	67.1	40.0
Hindi	26.8	21.5	65.0	44.4	28.8	16.0	70.7	45.7	21.2	10.6
Hungarian	24.3	17.0	31.2	18.1	30.7	18.4	40.3	18.3	30.7	18.0
Icelandic	49.9	38.2	42.9	18.3	42.3	19.9	53.6	20.5	45.3	18.6
Indonesian	10.2	7.1	25.5	11.7	23.8	12.1	31.9	11.6	23.4	11.8
Javanese	67.9	68.5	32.8	19.6	32.8	20.0	58.8	27.2	34.2	19.5
Kannada	77.7	37.0	18.8	14.4	15.8	12.9	41.3	25.2	17.7	13.3
Kazakh	48.8	37.7	30.2	17.4	30.2	17.7	63.8	19.5	31.6	17.4
Khmer	103.8	128.9	26.0	19.9	25.7	19.8	70.7	52.4	26.7	19.7
Korean	16.4	14.3	58.7	37.5	59.9	37.3	82.1	58.2	68.3	40.1
Lao	101.4	101.5	48.9	45.4	24.2	22.8	62.1	56.6	22.6	16.9
Latvian	32.0	23.1	20.8	12.0	20.9	12.1	24.5	11.9	21.8	12.1
Malay	12.2	8.7	25.3	12.3	25.9	13.2	32.4	12.1	26.1	12.5
Malayalam	101.1	100.7	23.7	19.1	19.5	16.6	39.1	25.6	20.4	15.3
Marathi	63.2	38.3	32.5	19.0	19.2	13.5	28.0	14.9	20.9	13.4
Mongolian	103.7	110.5	55.7	29.3	54.9	32.9	67.7	28.7	55.3	32.3
Persian	41.0	32.9	39.7	22.9	39.9	22.5	44.4	21.3	42.9	22.0
Polish	8.0	5.4	21.5	11.4	20.8	11.6	33.0	11.0	25.1	11.3
Portuguese	5.0	4.3	16.1	10.8	16.3	10.8	19.3	10.2	17.7	10.5
Punjabi	102.0	102.4	41.4	29.9	30.4	20.7	99.0	91.0	31.0	19.8
Romanian	20.0	14.4	27.9	18.8	28.4	19.1	31.3	17.8	27.4	18.3
Russian	7.2	5.6	30.3	14.6	30.4	14.3	38.8	14.7	35.0	15.0
Shona	143.9	121.0	38.1	30.4	37.7	30.1	43.0	29.9	37.8	29.6
Somali	104.0	102.9	51.8	42.8	52.5	43.0	54.5	42.9	53.8	42.8
Spanish	3.6	3.0	12.2	7.8	12.4	8.2	14.0	7.8	14.0	8.7
Swahili	52.8	39.3	22.9	16.0	23.3	15.6	29.6	16.8	23.7	16.0
Swedish	11.2	8.5	29.9	17.4	30.5	17.5	38.2	17.2	33.5	17.4
Tajik	74.0	85.8	59.8	46.6	33.9	19.2	59.0	39.5	25.7	15.7
Tamil	23.1	17.5	24.2	18.3	21.9	16.3	25.3	17.3	23.9	16.3
Telugu	82.8	99.0	19.4	13.7	19.6	13.7	24.5	15.8	22.1	13.6
Thai	15.4	11.5	18.2	13.6	18.1	13.6	27.6	18.8	20.7	14.3
Turkish	10.4	8.4	28.6	17.3	28.7	17.5	31.2	16.1	30.9	16.9
Ukrainian	11.6	8.6	31.1	13.6	31.7	13.5	39.2	13.3	33.3	13.6
Urdu	28.2	22.6	42.3	22.7	33.1	20.1	46.4	25.1	36.9	20.5
Vietnamese	12.7	10.3	44.5	18.6	47.5	20.7	56.6	21.0	52.9	19.8
Welsh	40.8	33.0	48.9	20.8	49.0	20.8	54.9	21.4	51.4	20.9
Yoruba	105.1	94.8	61.2	49.7	61.9	49.4	62.7	50.2	64.2	49.4
	50.1	44.3	33.3	20.7	31.0	19.1	44.2	24.8	32.5	18.7

Table A1: **Comparison to Whisper on the FLEURS test set.** We report WER for each of the 54 languages supported by both MMS and Whisper, except for Thai (tha), Lao (lao), Burmese (mya) and Khmer (khm) where we report CER. We apply Whisper normalization for both reference and hypothesis for measuring CER/WER.

	#lang	lbld train data (h)	FLEURS-54 (M)	 dev	 test
Prior Work					
Whisper medium	99	680K	769M	-	50.1
Whisper large-v2	99		1,550M	-	44.3
This Work					
MMS	61	3K	965M	33.6	33.3
+ CC LM	61	3K	965M	20.9	20.7
MMS (LSAH)	61	3K	1,096M	31.4	31.0
+ CC LM	61	3K	1,096M	19.1	19.0
MMS	1,107	45K	965M	44.7	44.2
+ CC LM	1,107	45K	965M	24.8	24.8
MMS (LSAH)	1,107	45K	3,346M	32.8	32.5
+ CC LM	1,107	45K	3,346M	18.7	18.7

Table A2: **Comparison to Whisper.** We report average WER on the 54 languages of the FLEURS benchmark supported by both Whisper and MMS (FLEURS-54). MMS is a CTC-based model and to enable a fairer comparison we use n-gram models trained on web data when comparing to Whisper whose decoder is a neural sequence-model and serves as a language model that was trained on billions of web tokens.

Appendix D. Comparison to USM

	Dev CER	Test CER	LM
Afrikaans	6.3	6.3	CC LM
Amharic	6.8	7.0	CC LM
Arabic	4.3	4.9	CC LM
Assamese	7.3	7.7	CC LM
Asturian	5.2	5.1	CC LM
Azerbaijani	4.2	3.9	CC LM
Belarusian	3.8	3.8	CC LM
Bengali	5.1	5.3	CC LM
Bosnian	3.8	3.4	FL LM
Bulgarian	3.0	3.2	CC LM
Catalan	2.8	2.9	CC LM
Cebuano	3.6	4.4	CC LM
Czech	3.3	3.0	CC LM
Sorani	6.5	7.4	FL LM
Mandarin	14.8	14.9	CC LM
Welsh	5.7	5.9	CC LM
Danish	5.5	5.7	CC LM
German	3.0	2.7	CC LM
Greek	4.0	3.7	CC LM
English	4.6	4.3	CC LM
Estonian	2.9	2.7	FL LM
Persian	4.0	4.1	FL LM
Finnish	2.6	2.4	FL LM
French	4.1	4.1	CC LM
Fula	13.9	13.8	FL LM
Irish	19.3	19.5	CC LM
Galician	2.8	2.7	CC LM
Gujarati	5.2	5.1	CC LM
Hausa	5.5	5.9	CC LM
Hebrew	15.5	12.9	CC LM
Hindi	5.4	4.7	CC LM
Croatian	3.5	3.2	CC LM
Hungarian	4.1	4.2	CC LM
Armenian	3.2	3.2	FL LM
Igbo	10.7	11.1	FL LM
Indonesian	2.3	2.1	CC LM
Icelandic	5.2	4.1	CC LM
Italian	1.5	1.4	CC LM
Javanese	4.3	3.8	CC LM
Japanese	14.5	14.6	CC LM
Kamba	12.5	11.3	FL LM
Kannada	4.6	4.4	FL LM
Georgian	3.4	3.6	CC LM
Kazakh	2.9	3.1	CC LM
Kabuverdianu	4.3	4.3	FL LM
Khmer	9.9	10.3	CC LM
Kyrgyz	4.1	3.6	FL LM
Korean	11.4	11.8	CC LM
Lao	26.5	24.1	CC LM
Latvian	2.8	2.2	CC LM
Lingala	4.0	4.3	FL LM
Lithuanian	4.3	3.7	CC LM
Luxembourgish	8.0	7.5	CC LM
Ganda	8.0	8.4	FL LM
Luo	4.9	5.0	FL LM
Malayalam	4.4	4.2	FL LM
Marathi	7.2	6.6	CC LM
Macedonian	2.0	1.9	CC LM
Maltese	3.7	3.7	CC LM
Mongolian	5.6	5.9	CC LM
Maori	5.9	6.9	CC LM
Burmese	8.9	9.2	CC LM
Dutch	3.7	3.1	CC LM
Norwegian	3.7	4.1	CC LM
Nepali	8.3	7.7	CC LM
Northern	6.8	6.1	FL LM
Nyanja	6.3	6.7	CC LM
Occitan	7.8	8.2	CC LM
Oromo	15.7	16.2	CC LM
Oriya	6.3	7.1	CC LM
Punjabi	7.4	7.0	CC LM
Polish	2.6	2.7	CC LM
Portuguese	2.8	2.8	CC LM
Pashto	13.1	14.1	CC LM
Romanian	3.7	3.1	CC LM
Russian	3.1	3.0	CC LM
Slovak	2.7	2.5	CC LM
Slovenian	4.0	3.7	CC LM
Shona	3.8	4.1	FL LM
Sindhi	7.2	7.2	FL LM
Somali	12.8	13.0	CC LM
Spanish	1.8	2.1	CC LM
Serbian	10.3	12.7	FL LM
Swedish	4.7	4.6	CC LM
Swahili	3.3	3.4	CC LM
Tamil	8.2	9.0	FL LM
Telugu	6.6	6.9	CC LM
Tajik	4.0	4.5	CC LM
Filipino	3.1	3.1	CC LM
Thai	7.6	8.3	CC LM
Turkish	3.5	3.1	CC LM
Ukrainian	3.3	2.9	CC LM
Umbundu	10.7	10.2	FL LM
Urdu	9.8	8.1	CC LM
Uzbek	4.7	5.0	CC LM
Vietnamese	5.9	6.1	CC LM
Wolof	11.2	11.5	FL LM
Xhosa	6.0	6.1	FL LM
Yoruba	17.0	16.1	FL LM
Cantonese	12.9	12.4	CC LM
Malay	2.8	2.6	CC LM
Zulu	5.2	5.5	FL LM
Average	6.3	6.3	

Table A3: **Results on FLEURS-102.** We show character error rate on the dev and test sets of all 102 FLEURS languages for MMS when fine-tuned on the labeled data of FLEURS using language-specific adapters/heads. For inference we choose between two n-gram language models (LM) based on dev set accuracy: a word-based model trained on Common Crawl (CC LM) or a character-based model trained on the FLEURS training transcriptions (FL LM).

Appendix E. Gender Bias Study

	Num Samples		MMS ASR CER ↓			FLEURS ASR CER ↓		
	Female	Male	Female	Male	Overall	Female	Male	Overall
Assamese	139	279	15.3	15.1	15.4	8.6	8.0	8.9
Bulgarian	189	206	9.0	10.2	7.9	3.6	3.5	3.6
Welsh	150	296	15.8	19.4	14.1	8.4	11.5	6.9
Greek	125	146	11.4	9.3	13.1	5.4	4.0	6.6
English	245	149	10.8	10.7	10.9	6.0	5.5	6.8
Spanish	130	278	5.2	5.8	5.0	2.3	2.9	2.0
Fula	107	166	27.2	29.0	26.2	14.8	21.4	11.0
Finnish	352	63	7.6	7.1	10.2	2.7	2.6	3.8
French	62	227	9.1	10.0	8.8	5.4	6.4	5.2
Gujarati	149	283	13.3	12.7	13.7	6.2	6.2	6.2
Hindi	120	119	15.1	15.5	14.7	6.6	8.3	5.1
Kazakh	136	232	7.8	7.4	8.0	3.1	2.4	3.6
Khmer	98	228	26.1	22.8	27.6	14.1	11.2	15.4
Kannada	245	123	10.0	10.4	9.3	5.2	5.2	5.1
Korean	93	133	24.3	26.0	22.9	13.5	14.0	13.1
Kyrgyz	273	149	10.6	12.1	7.9	4.5	5.7	2.1
Ganda	228	78	13.4	13.3	13.6	8.2	7.9	9.0
Latvian	177	179	7.0	6.6	7.3	3.0	2.7	3.3
Marathi	148	295	13.2	9.3	15.0	8.2	4.8	9.9
Polish	95	243	6.6	7.2	6.3	3.3	3.8	3.1
Russian	203	153	8.0	7.4	8.9	4.0	3.7	4.6
Shona	118	275	10.5	8.6	11.4	4.1	2.2	5.1
Swedish	64	266	10.1	8.5	10.5	5.6	4.6	5.9
Swahili	59	152	7.7	11.1	6.5	3.7	3.9	3.6
Tamil	214	163	15.3	17.0	13.2	8.9	10.1	7.4
Telugu	76	235	12.7	11.3	13.1	7.9	8.2	7.8
Tajik	116	123	11.1	12.5	9.9	4.3	4.9	3.7
Average	-	-	12.4	12.3	12.4	6.5	6.2	6.4

Table A4: **Analysis of Gender Bias.** We compare ASR models trained on MMS-lab data and FLEURS data. We report dev CER per gender of the speakers for 27 languages of FLEURS for which MMS-lab provides data and for which there are at least 50 samples for each gender.