Edge-Optimized Offline Voice Assistants for Low-Resource Languages: Tokenization and Deployment Insights from Garo

Badal Nyalang
MWire Labs, Meghalaya, India
nyalang@mwirelabs.com

Abstract

Deploying voice assistants for low-resource languages on edge hardware requires overcoming simultaneous constraints of model efficiency, offline operation, and limited compute. We report on a fully offline educational voice assistant for Garo, a low-resource Tibeto-Burman language spoken by approximately 1–1.2 million people in Northeast India, targeting visually impaired users in low-connectivity settings. Through systematic component optimization, we surface three actionable efficiency insights: (1) a custom subword vocabulary extension yields a 34% reduction in Garo token sequence length, directly enabling on-device inference with a 360M parameter language model and saving approximately 1.5–2 seconds per response at CPU inference speeds; (2) continued pretraining on 55,623 Garo sentences reduces perplexity 262× (from 8,336 to 31.8) with no degradation on English, demonstrating effective cross-lingual transfer; and (3) end-to-end offline operation is achievable on a 16 GB RAM x86 mini-PC at approximately 20 seconds total latency. Per-unit hardware cost is approximately INR 40,000–60,000 (USD $480–720). These findings provide concrete guidance for practitioners building efficient offline NLP systems in resource-constrained, low-resource language settings.

1 Introduction

Voice-based interfaces are widely presented as a solution for accessibility and digital inclusion. In practice, most deployed voice assistants assume continuous internet connectivity, cloud-based inference, and interaction through smartphones or smart displays. These assumptions break down for speakers of low-resource languages, particularly visually impaired users in regions with unreliable connectivity and limited access to assistive technology (Joshi et al., 2020; Ahia et al., 2021).

Garo is a Tibeto-Burman language spoken by approximately 1–1.2 million people in Northeast India and neighboring regions. Despite this speaker population, Garo remains technologically underserved: it is absent from mainstream speech and language platforms, and there is no commercial-grade voice assistant that operates in the language. The problem is compounded for visually impaired users, who must rely on audio-only interaction and often require sighted assistance to configure or operate smartphone-based systems. In rural and semi-rural settings, intermittent connectivity and device cost further restrict access.

Deploying language technology on edge hardware for low-resource languages presents a distinct efficiency challenge: models must fit within strict memory and compute budgets, operate without cloud connectivity, and serve languages for which large pretrained models provide little direct coverage. Standard efficiency approaches such as quantization alone, or simply using a smaller model, are insufficient without language-specific adaptation. A base model trained on high-resource languages will exhibit near-random perplexity on an unseen agglutinative language, and a standard tokenizer will produce sequence lengths that make even small models impractical on edge hardware.

In this work, we present a fully offline educational voice assistant for Garo that runs entirely on low-cost edge hardware, requires no smartphone or internet connectivity, and is operated through a single physical power interface followed by voice-only interaction. The system integrates Garo ASR, a Garo-adapted SmolLM2-360M language model with a custom tokenizer, and Garo TTS. Educational content up to middle-school level serves as a deliberately bounded domain, enabling predictable behavior and reducing the risk of erroneous or misleading responses in an assistive setting.

We position this work as a system and deployment contribution. Rather than proposing new modeling techniques, we focus on the efficiency findings and design trade-offs involved in integrating

existing low-resource language technologies into a deployable, offline, accessibility-first system. Our contributions are:

1. Empirical demonstration that custom subword vocabulary extension reduces Garo token sequence length by 34%, enabling edge LM deployment with a 360M parameter model within a 16 GB RAM budget.

2. Quantification of continued pretraining (CPT) effectiveness: a 262× perplexity improvement on Garo using only 55,623 sentences, with cross-lingual transfer preserving English fluency via a replay strategy.

3. End-to-end latency characterization across all pipeline components, and a concrete hardware cost analysis for offline accessibility-first deployment in low-resource settings.

2 Related Work

2.1 Low-Resource Speech and Language Processing

Low-resource speech and language technologies commonly rely on transfer learning and multilingual pretraining to reduce data requirements (Besacier et al., 2014; Pratap et al., 2024). For ASR specifically, recent results demonstrate that offline recognition is achievable with tens of hours of data (Radford et al., 2023). However, most studies evaluate components in isolation and rarely address deployment constraints such as offline operation, latency, or memory limits.

Language models for low-resource and indigenous languages are often adapted through continued pretraining, vocabulary extension, and instruction tuning (Gururangan et al., 2020; Chung et al., 2024). Custom tokenization has been shown to substantially improve efficiency for languages with complex morphology or non-standard orthography (Gutierrez-Vasques et al., 2023; Raj et al., 2025). Tokenizer design also introduces measurable unfairness between languages, with low-resource languages typically requiring more tokens per word than high-resource ones (Petrov et al., 2023). Despite these advances, most work assumes cloud-based inference and does not consider feasibility on edge devices with strict resource constraints. For Garo specifically, Nyalang (2026) introduced the first multilingual BERT-style model covering Latin-script Garo, establishing a foundation for downstream NLP tasks in the language.

2.2 Offline and Edge NLP Systems

Prior work on offline and edge NLP explores model compression, quantization, and domain restriction to enable on-device inference. Recent work demonstrates INT4/INT8 quantization achieving 2–4× speedups with minimal accuracy loss for edge LLM deployment (Lin et al., 2025). Dutta et al. (2025) adapted Whisper for lightweight edge ASR targeting children's speech, demonstrating feasibility of domain-specific optimization for resource-constrained deployment. Xu et al. (2024) achieved conformer-based speech recognition on extreme edge computing hardware through architectural optimizations. **?** demonstrated edge TTS deployment for Te Reo Māori and 'Ōlelo Hawaii, showing that indigenous language speech synthesis is feasible on low-cost devices. Di Leo et al. (2025) further demonstrated ultra-low latency speech-to-text on edge hardware, providing a practical reference point for end-to-end latency budgets in offline systems.

While these approaches show promise, they typically focus on single components rather than complete spoken interaction systems. End-to-end offline voice assistants remain rare, particularly for low-resource languages and accessibility-driven use cases. Miranda et al. (2026) survey the intersection of multilinguality and edge deployment for the Global South, identifying end-to-end offline systems as a critical gap in the literature; our work addresses this gap directly for Garo.

2.3 Voice Interfaces and Accessibility

Voice interfaces are widely recognized as important accessibility tools for visually impaired users. However, most deployed assistants rely on smartphones or cloud-backed devices, which introduce barriers related to visual setup, connectivity, and reliability. Recent assistive technologies for blind and low-vision users often combine voice interfaces with smartphone-based navigation or object recognition systems (Ainary, 2025). These systems assume reliable connectivity, touchscreen interaction for configuration, and high-resource language support, limiting applicability in low-resource language settings with intermittent connectivity. Our work differs by targeting fully offline operation and eliminating smartphone dependency to address accessibility barriers in rural, low-connectivity environments.

3 System Overview

3.1 Scope and Design Constraints

The system is a fully offline educational voice assistant for Garo, designed for audio-only interaction and edge deployment. It targets primary and middle-school educational content as a bounded domain. All components run locally without cloud connectivity or smartphone dependency. Design constraints are:

- No internet connectivity
- No smartphone or screen-based interaction
- Audio-only feedback
- Domain-restricted language generation
- Predictable behavior prioritized over open-ended conversation

3.2 Pipeline Components

Speech Recognition (ASR). The system uses a publicly released Garo ASR model developed by the authors. The model achieves 9.2% Word Error Rate on held-out test data and supports fully offline inference. It has been validated through deployment to over 250 distinct users on HuggingFace Spaces. The model is suitable for short educational queries and command-style interaction.

Language Model. Language understanding and response generation are handled by SmolLM2-360M (allal et al., 2025), adapted to Garo via continued pretraining (CPT) on 55,623 Garo sentences (1.2M tokens). A 15,000-sentence English replay set (0.2M tokens) is included to mitigate catastrophic forgetting (Gururangan et al., 2020), following standard causal language modeling with the extended tokenizer. The model uses a 2,048-token context window and currently runs in half precision (fp16). Domain restriction is enforced at two levels: training data (supervised fine-tuning uses only curriculum-aligned educational Garo Q&A pairs) and inference (prompt-level system instructions anchor the model to curriculum scope). Out-of-domain queries trigger a fixed confidence-based fallback response rather than free generation, prioritizing reliability and user trust over conversational fluency.

Tokenizer. The base SmolLM2 tokenizer (49,152 tokens) is extended with Garo-specific subword units, expanding vocabulary to 55,923 tokens (+6,771 units). This reduces average Garo token sequence length from 23.5 to 15.5 tokens per sentence, a 34% reduction that directly lowers autoregressive decoding steps and saves approximately 1.5–2 seconds per response at current CPU inference speeds. English sequence length increases from 9.5 to 15.5 tokens per sentence due to vocabulary expansion with Garo-specific units; the tokenizer may split English words into more fragments when Garo-optimized units are not applicable. This trade-off is acceptable as the system prioritizes Garo efficiency for the target educational domain.

Text-to-Speech (TTS). Speech synthesis is trained using the same Garo speech data employed for ASR, with additional focused recordings to improve naturalness and intelligibility. TTS output is optimized for extended listening in educational settings and runs fully offline on-device.

3.3 Interaction Pipeline

1. User powers on the device via a single physical button.
2. The system activates and begins listening. Wake-word detection using lightweight keyword spotting is planned for production deployment; the current prototype uses continuous listening.
3. User speaks a query in Garo.
4. ASR transcribes speech locally.
5. The language model generates a domain-constrained response.
6. TTS renders the response as audio.

All processing occurs on-device. When ASR confidence is low, the system requests repetition rather than guessing. Correctness and user trust are prioritized over conversational fluency.

3.4 Hardware

The system targets low-cost x86 edge hardware such as the Lenovo ThinkCentre M93 (Intel i7-4th gen, 16 GB RAM) or equivalent compact desktop systems with 8–16 GB RAM. ASR and TTS operate fully offline on the same device.

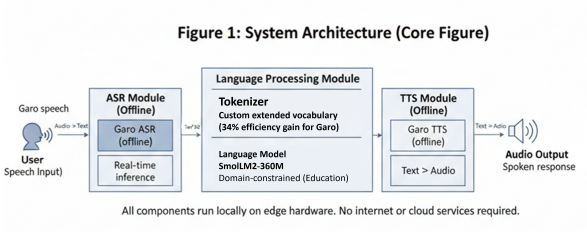

Figure 1: System architecture. All components run locally on edge hardware: Garo ASR, domain-constrained SmolLM2-360M with a custom tokenizer optimized for Garo, and Garo TTS. No internet or cloud services are required.

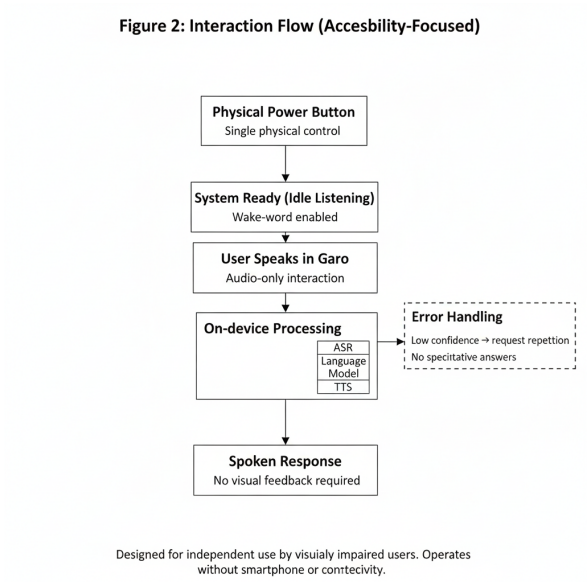

Figure 2: Accessibility-focused interaction flow. A single physical power button initiates audio-only interaction, enabling independent use by visually impaired users without smartphones or connectivity.

4 Efficiency Analysis

4.1 Tokenization Efficiency

Table 1 shows the effect of vocabulary extension on sequence length. The 34% reduction in Garo token count directly reduces inference cost for the autoregressive language model. For a 20-token educational response, this translates to approximately 1.5–2 seconds of saved CPU decoding time per query, representing the difference between a system that fits within an interactive latency budget and one that does not.

English sequence length increases as a side effect of adding Garo-specific subword units. This trade-off is acceptable given that the target domain is Garo educational content.

Language	Base	Extended
Garo (tokens/sent.)	23.5	15.5
English (tokens/sent.)	9.5	15.5

Table 1: Average token sequence length per sentence before and after extending the tokenizer with Garo-specific subword units. The 34% reduction in Garo sequence length directly lowers on-device inference cost.

Language	Base PPL	After CPT
Garo	8,336.2	31.8
English	792.9	31.9

Table 2: Perplexity (PPL) before and after continued pretraining (CPT) on Garo data, evaluated on held-out sets from the same domain. Base PPL reflects the original SmolLM2-360M checkpoint with the extended tokenizer but before CPT. CPT reduces Garo perplexity 262×. English perplexity is measured on the same English replay set across both conditions, confirming no catastrophic forgetting.

4.2 Continued Pretraining Effectiveness

Table 2 shows perplexity before and after CPT. The base SmolLM2-360M model has near-random perplexity on Garo (8,336.2) due to the complete absence of Garo in its pretraining data. CPT on 55,623 Garo sentences reduces this 262× to 31.8, a level consistent with functional language generation for a bounded educational domain.

English perplexity also improves substantially (from 792.9 to 31.9), demonstrating effective cross-lingual transfer. The 15,000-sentence English replay strategy successfully prevents catastrophic forgetting while enabling strong Garo adaptation. This result suggests that the data barrier for low-resource language model adaptation is lower than commonly assumed: 55,623 sentences suffice for a 262× perplexity reduction on a completely unseen agglutinative language, consistent with findings in low-resource ASR showing strong gains from modest-scale continued pretraining (Mutisya and Mugane, 2026).

4.3 End-to-End Latency

Table 3 summarizes component-level latency on CPU-only baselines. ASR transcription of a 5–7 second Garo utterance takes approximately 13 seconds on HuggingFace Spaces (2 vCPU, 16 GB RAM) using an unoptimized Whisper-based prototype. The production ASR model, which achieves 9.2% WER on held-out Garo test data, was evaluated separately from this latency measurement; on-

Comp.	Latency	Mem.	Env.
ASR	∼13s	∼4 GB	HF Spaces
LM	∼5s	2–3 GB	Colab CPU
TTS	∼2–3s	–	Est.
Total	**∼20s**	**<16 GB**	–

Table 3: Component-level latency and peak memory on CPU-only baselines. All components fit within a 16 GB RAM budget. TTS latency is an estimate; on-device benchmarks on target hardware remain future work.

device latency for the production model remains future work. LM generation of a 20-token educational response takes approximately 5 seconds on a CPU-only environment (Google Colab, 12.7 GB RAM). TTS synthesis adds approximately 2–3 seconds. Total end-to-end latency is approximately 20 seconds.

While not suitable for real-time conversation, this latency is comparable to a teacher's verbal response time and is acceptable for educational Q&A scenarios where correctness is prioritized over speed. These measurements represent CPU feasibility baselines; direct on-device benchmarks on the target hardware remain future work.

5 Accessibility-First Design

The system is designed explicitly for visually impaired users in low-connectivity, low-resource settings. Accessibility constraints directly shape system architecture and interaction design.

No smartphone interface. Smartphone-based assistants require visual navigation for setup, error recovery, and application management. These steps often require sighted assistance and reliable connectivity. The system avoids mobile applications entirely.

Physical control and audio-only interaction. A single physical power button enables independent use. Once powered on, all interaction is speech-based. No screens, menus, or visual indicators are required.

Offline operation. All components run locally to ensure reliability in environments with intermittent or no internet access. Offline processing also avoids external data transmission, supporting user privacy.

Domain restriction for safety. Language generation is constrained to educational content. This reduces hallucination risk and ensures predictable, curriculum-aligned responses suitable for assistive use. In this setting, domain restriction is not a limitation but an enabler: it allows a 360M parameter model to serve reliably in a use case that would otherwise require much larger cloud-based systems.

Conservative error handling. When ASR confidence is low, the system requests repetition rather than guessing. Correctness and user trust are prioritized over conversational fluency.

6 Deployment Readiness and Cost

Hardware and cost. Per-unit hardware cost is approximately INR 40,000–60,000 (USD $480–720). Smartphone-based solutions appear cost-effective at INR 8,000–12,000, but present significant barriers in this specific context: in rural, low-connectivity settings, visually impaired users may face difficulties independently navigating touch interfaces, managing applications, or recovering from errors without sighted assistance. The 5× cost premium for dedicated hardware is justified by enabling genuine user autonomy. Power infrastructure (battery backup or solar panels) adds approximately INR 10,000–15,000 per unit to address unreliable electricity in rural deployment sites.

Deployment strategy. Individual purchase is not the target model. Community center placement (shared access across multiple schools) or government procurement programs are the viable pathways, consistent with the hardware cost profile and the rural deployment context.

Prototype status. The system is deployed as a working offline prototype with physical controls and audio-only interaction. End-to-end voice interaction is functional without external services. The Garo ASR component has been validated through deployment to over 250 distinct users on HuggingFace Spaces, indicating real-world usage beyond internal testing.

7 Discussion

Three efficiency lessons emerge from this work.

First, vocabulary extension is a prerequisite for edge LM deployment on agglutinative low-resource languages, not merely an optimization. A standard tokenizer applied to Garo produces sequence lengths that make even small models impractical on CPU hardware. The 34% reduction achieved here was the primary enabler for fitting

SmolLM2-360M within a 16 GB interactive budget.

Second, continued pretraining is highly effective at small scale. Only 55,623 sentences suffice for a 262× perplexity reduction on a language completely absent from pretraining data. Practitioners working on similarly underserved agglutinative languages should not assume large data volumes are required; in this setup, modest corpora were sufficient for meaningful adaptation when combined with appropriate tokenization and a small replay set to prevent catastrophic forgetting. These findings are specific to Garo, SmolLM2-360M, and this training configuration, and may not generalize without further validation.

Third, domain restriction is an efficiency multiplier in assistive settings. Constraining generation to educational content allows a 360M parameter model to serve reliably in a use case that would otherwise require much larger cloud-based systems. The combination of tokenization efficiency, CPT, and domain restriction is what makes offline edge deployment feasible at this hardware cost point.

7.1 Sustainability and Environmental Impact

A key advantage of the fully offline, edge-based design is its substantially lower environmental footprint compared to cloud-dependent voice assistants. Continuous cloud inference for even modest deployment scales incurs significant energy costs from data centers, networking, and always-on servers. In contrast, the system runs entirely locally on low-power x86 mini-PCs, eliminating ongoing cloud compute and data transfer emissions. Eliminating cloud dependency removes the associated network and data center energy costs entirely, a meaningful reduction for community-scale deployment (Miranda et al., 2026).

Domain restriction to educational content further amplifies efficiency: a 360M parameter model suffices for reliable performance, avoiding the energy demands of larger general-purpose LLMs. This approach aligns with broader goals of sustainable AI deployment in the Global South, making high-quality assistive technology accessible without exacerbating environmental costs or relying on unreliable infrastructure (Miranda et al., 2026). Future work will include detailed power profiling and carbon footprint comparisons on target hardware.

Limitations

This system is designed for a bounded educational domain and does not support open-ended conversational use. Language generation quality is constrained by the use of a 360M parameter model, which limits expressiveness compared to larger cloud-based systems.

Latency measurements are from cloud CPU baselines (HuggingFace Spaces, Google Colab) rather than the target edge hardware (Lenovo ThinkCentre M93). On-device benchmarks with end-to-end throughput analysis remain future work. Systematic INT8/INT4 quantization experiments to reduce LM latency are planned but not yet conducted. TTS quality metrics (MOS, intelligibility) have not been formally evaluated.

Controlled user studies with visually impaired participants are needed to validate usability and latency assumptions. The 250+ ASR demo users represent general usage, not the target blind and low-vision population. Performance degrades in high-noise environments and for dialectal variation not fully covered by available speech data. Wake-word detection, energy consumption, and thermal constraints on edge hardware are implementation details not yet characterized. Ablation studies isolating the contribution of tokenizer extension alone, CPT alone, and the combined system, including perplexity, token length, and latency for each setting remain important future work for establishing the relative contribution of each component.

Acknowledgments

The author thanks the Garo-speaking community for their assistance in data collection. Data collection and compute resources were supported by MWire Labs.

References

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. The low-resource double bind: An empirical study of pruning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333.

Bhanuja Ainary. 2025. Audo-sight: Enabling ambient interaction for blind and visually impaired individuals. *arXiv preprint arXiv:2505.00153*.

Loubna Ben allal, Anton Lozhkov, Elie Bakouch, Gabriel Martin Blazquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner,

Caleb Fahlgren, Xuan Son NGUYEN, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, and 3 others. 2025. SmolLM2: When smol goes big — data-centric training of a fully open small language model. In *Second Conference on Language Modeling*.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. 56:85–100.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Stefano Di Leo, Luca De Cicco, and Saverio Mascolo. 2025. Real-time speech-to-text on edge: A prototype system for ultra-low latency communication with AI-powered NLP. *Information*, 16(8):685.

Satwik Dutta, Shruthigna Chandupatla, and John H. L. Hansen. 2025. Adapting whisper for lightweight and efficient automatic speech recognition of children for on-device edge applications. In *Proceedings of the Workshop on Child Computer Interaction (WOCCI)*.

Suchin Gururangan, Ana Marasović, Swabha Swayamditta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Ximena Gutierrez-Vasques, Christian Bentz, and Tanja Samardžić. 2023. Languages through the looking glass of bpe compression. *Computational Linguistics*, 49(4):943–1001.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2025. QServe: W4A8KV4 quantization and system co-design for efficient LLM serving. In *Proceedings of Machine Learning and Systems*, volume 7.

Lester James V. Miranda, Songbo Hu, Roi Reichart, and Anna Korhonen. 2026. Multilinguality at the edge: Developing language models for the Global South. *arXiv preprint arXiv:2604.21637*.

Hillary Mutisya and John Mugane. 2026. Continued pretraining for low-resource Swahili ASR: Achieving state-of-the-art performance with minimal labeled data. *arXiv preprint arXiv:2603.11378*.

Badal Nyalang. 2026. NE-BERT: A multilingual language model for nine Northeast Indian languages. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 1–12, Rabat, Morocco. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Bharath S Raj, Garvit Suri, Vikrant Dewangan, and Raghav Sonavane. 2025. When every token counts: Optimal segmentation for low-resource language models. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages (LoResLM 2025)*.

Mingbin Xu, Alex Jin, Sicheng Wang, Mu Su, Tim Ng, Henry Mason, Shiyi Han, Zhihong Lei, Yaqiao Deng, Zhen Huang, and Mahesh Krishnamoorthy. 2024. Conformer-based speech recognition on extreme edge-computing devices. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, page 131–139.