Evoxtral: Expressive Tagged Transcription via Supervised Fine-Tuning and Rejection Sampling

Yongkang Zou[1]

[1]Mistral AI Online Hackathon 2026 — W&B Fine-Tuning Track

Abstract

Standard automatic speech recognition (ASR) systems discard paralinguistic and expressive information present in spoken audio, producing plain text that fails to capture sighs, laughs, hesitations, emotional tone, and other prosodic cues. We present **Evoxtral**, a LoRA adapter for `Voxtral-Mini-3B-2507` Mistral AI [2025] that produces transcriptions enriched with inline expressive audio tags drawn from the ElevenLabs v3 tag vocabulary ElevenLabs [2025]. We apply a two-stage post-training pipeline: supervised fine-tuning (SFT) followed by rejection sampling fine-tuning (RAFT) Yuan et al. [2023]. SFT reduces word error rate (WER) by 33% relative (6.64% → 4.47%) and increases Tag F1 from 22.0% to 67.2%. The subsequent RAFT stage further improves Tag F1 to 69.4% and Tag Recall to 72.7% at a marginal WER cost. We release two model variants, a serverless inference API, and a live interactive demo.

1 Introduction

Modern ASR pipelines excel at converting speech to text with low word error rates, yet they systematically strip out the expressive dimension of human communication. When a speaker sighs before a sentence, laughs nervously mid-phrase, or whispers for emphasis, these paralinguistic signals carry meaning that plain transcription cannot represent. This information is especially critical for downstream text-to-speech (TTS) synthesis: next-generation TTS systems such as ElevenLabs v3 ElevenLabs [2025] consume inline expressive tags to control prosody, affect, and delivery at a fine-grained level.

We ask: *can a multimodal audio-language model be trained to produce ASR output that preserves expressive content through inline tags?* We answer affirmatively with **Evoxtral**, built on Voxtral-Mini-3B-2507 Mistral AI [2025].

Why Voxtral. We chose Voxtral-Mini-3B as our base model for several reasons. First, as a *generative* audio-language model (rather than a CTC or encoder-only ASR), Voxtral decodes transcriptions autoregressively—meaning it can naturally produce arbitrary inline tokens such as `[laughs]` or `[nervous]` within the text stream, without requiring architectural changes. Traditional ASR models constrain their output vocabulary to words and punctuation; Voxtral's LLM decoder has no such limitation. Second, Voxtral's compact 3B-parameter architecture makes LoRA fine-tuning feasible on a single A10G GPU within hackathon time constraints, while still delivering competitive ASR quality. Third, Voxtral was itself trained with post-training alignment (SFT + DPO + Online DPO) Mistral AI [2025], meaning the model is already instruction-following and amenable to further fine-tuning—a strong foundation for adding new capabilities. Finally, as a Mistral AI model released under Apache 2.0, Voxtral aligns with the hackathon's focus on the Mistral ecosystem and enables open redistribution of our adapters.

Our approach fine-tunes Voxtral-Mini-3B-2507 using parameter-efficient LoRA adapters Hu et al. [2022] on a synthetically generated dataset of expressive speech paired with tagged transcriptions.

To illustrate the contrast, consider the following example:

> **Standard ASR:** "So I was thinking maybe we could try that new restaurant downtown. I mean if you're free this weekend."
>
> **Evoxtral:** "`[nervous]` So... `[stammers]` I was thinking maybe we could... `[clears throat]` try that new restaurant downtown? `[laughs nervously]` I mean, if you're free this weekend?"

The Evoxtral output captures hesitation, nervous laughter, and a throat clear—paralinguistic content that is acoustically present but conventionally discarded. We make the following contributions:

1. A synthetic dataset of 1,010 expressive speech samples paired with tagged transcriptions across 17 ElevenLabs v3 tag types.
2. A two-stage fine-tuning recipe (SFT → RAFT) for expressive ASR using LoRA on Voxtral-Mini-3B.
3. A custom evaluation benchmark, **Evoxtral-Bench**, with seven metrics covering both transcription accuracy and tag generation quality.
4. Two released model variants optimized for different use cases: accuracy-critical (SFT) and expressiveness-critical (RL).

2 Related Work

Voxtral Mistral AI [2025]. Voxtral-Mini-3B-2507 is a multimodal audio-language model released by Mistral AI. It is built on a Whisper-based audio encoder fused with a Mistral language model backbone, trained via SFT, DPO, and online DPO. Our work builds directly on this foundation, adding expressive tagging capability via LoRA adaptation.

Reinforcement Learning for LLM-based ASR Shi et al. [2025]. Shi et al. apply group relative policy optimization (GRPO) to LLM-based ASR, achieving an 18% relative WER reduction without paired preference data. Closely related work [2025a,b,c] further demonstrates that RL-based training consistently outperforms SFT alone for speech understanding tasks. Our RAFT stage is philosophically aligned with this line of work but uses rule-based rejection sampling rather than policy gradient methods, making it simpler to implement and more stable to train.

LoRA Hu et al. [2022]. Low-rank adaptation inserts trainable low-rank matrices into the attention projections and feed-forward layers of a frozen base model, reducing the number of trainable parameters by orders of magnitude while matching full fine-tuning performance. We use LoRA with rank 64 and alpha 128, implemented via HuggingFace PEFT Mangrulkar et al. [2022].

Rejection Sampling Fine-Tuning Yuan et al. [2023]. Yuan et al. propose generating multiple model completions for each training input, scoring them with a reward function, and performing SFT on the highest-scoring completions. This approach is computationally simpler than policy gradient methods while still providing a reinforcement learning signal.

NEFTune Jain et al. [2024]. Jain et al. demonstrate that adding uniform random noise to embedding vectors during training improves instruction-following performance. We apply NEFTune with noise alpha = 5.0 during SFT to regularize training on our small dataset.

ElevenLabs v3 Audio Tags ElevenLabs [2025]. ElevenLabs v3 TTS introduces a structured vocabulary of inline expressive tags that control how synthesized speech is delivered. These tags—such as `[sighs]`, `[laughs]`, `[whispers]`, and `[nervous]`—are the target output vocabulary for Evoxtral.

ASR Evaluation JiWER [2024], Morris et al. [2004]. We compute WER and CER using the `jiwer` library JiWER [2024], following standard definitions from Morris et al. Morris et al. [2004].

3 Method

3.1 Dataset

We construct a synthetic dataset of 1,010 audio samples generated using the ElevenLabs TTS v3 API ElevenLabs [2025]. Each sample consists of a short spoken utterance (5–30 s) paired with a reference tagged transcription containing inline expressive tags. The dataset covers 17 tag types: `[sighs]`, `[laughs]`, `[whispers]`, `[nervous]`, `[frustrated]`, `[clears throat]`, `[pause]`, `[excited]`, `[stammers]`, `[gasps]`, `[sad]`, `[angry]`, `[calm]`, `[crying]`, `[shouts]`, `[confused]`, and `[scared]`.

The dataset is split into 808 training, 101 validation, and 101 test samples. Tag frequency follows a long-tail distribution: `[pause]` is the most common while `[confused]` and `[scared]` appear rarely. The Whisper-based audio encoder is kept frozen throughout training; only the language model backbone and multi-modal projector are fine-tuned.

3.2 Stage 1: Supervised Fine-Tuning (SFT)

We fine-tune Voxtral-Mini-3B-2507 Mistral AI [2025] using LoRA Hu et al. [2022] with the configuration shown in Table 1.

Table 1: SFT hyperparameters.

Hyperparameter	**Value**
LoRA rank	64
LoRA alpha	128
LoRA dropout	0.05
Target modules	`q/k/v/o_proj`, `gate/up/down_proj`, `mm_projector`
Learning rate	2×10^{-4}
LR schedule	Cosine decay
Epochs	3
Batch size	2 (effective 16 via gradient accumulation $\times 8$)
NEFTune noise alpha	5.0
Precision	bf16
Hardware	NVIDIA A10G (24 GB)
Training time	$\sim$25 minutes
Trainable parameters	124.8 M / 4.8 B (2.6%)

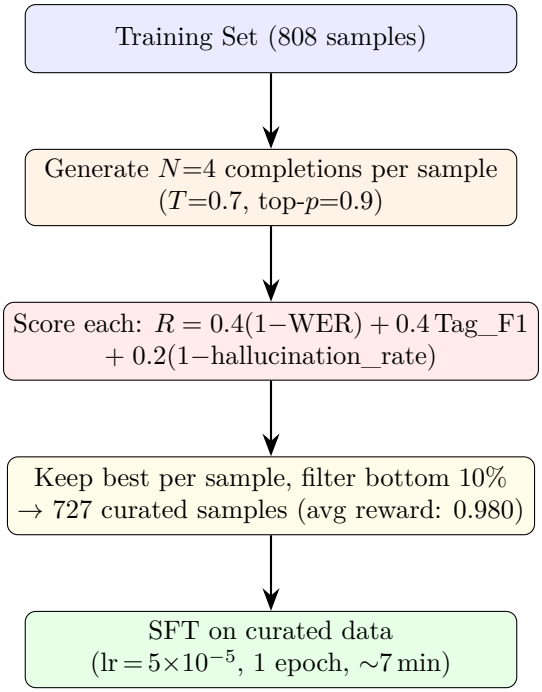

Figure 1: RAFT training pipeline. For each training sample, four completions are generated and scored by a rule-based reward function balancing transcription accuracy, tag quality, and hallucination avoidance. Only high-reward completions are retained for the final SFT pass.

The SFT objective is standard next-token prediction (cross-entropy) on the tagged transcription tokens, conditioned on the audio encoder outputs. NEFTune Jain et al. [2024] noise is applied to the input embeddings to reduce overfitting on the small training set.

3.3 Stage 2: Rejection Sampling Fine-Tuning (RAFT)

Following Yuan et al. Yuan et al. [2023] and inspired by Voxtral's own SFT→DPO recipe Mistral AI [2025] and GRPO-based ASR work Shi et al. [2025], we apply a rejection sampling stage to refine tag generation quality. The pipeline is illustrated in Figure 1.

The reward function balances three objectives: transcription accuracy (WER), tag generation quality (Tag F1), and avoidance of hallucinated tags:

$$R = 0.4 \times (1 - \text{WER}) + 0.4 \times \text{Tag\_F1} + 0.2 \times (1 - \text{Hallucination\_Rate}) \tag{1}$$

The 0.4/0.4/0.2 weighting reflects an equal priority on accuracy and expressiveness, with a penalty for hallucination. After filtering the bottom 10% by reward (threshold: $R > 0.954$), 727 of the original 808 training samples remain with a mean reward of 0.980. The RAFT SFT stage trains for one epoch at a reduced learning rate of 5×10^{-5}, completing in approximately 7 minutes with a final training loss of 0.021.

The full two-stage pipeline overview is shown in Figure 2.

4 Evaluation

4.1 Evoxtral-Bench

We evaluate on **Evoxtral-Bench**, a held-out benchmark of 50 test samples drawn from the 101-sample test split. We compute seven evaluation metrics:

- **WER** (Word Error Rate, ↓) — via `jiwer` JiWER [2024], tags stripped before comparison

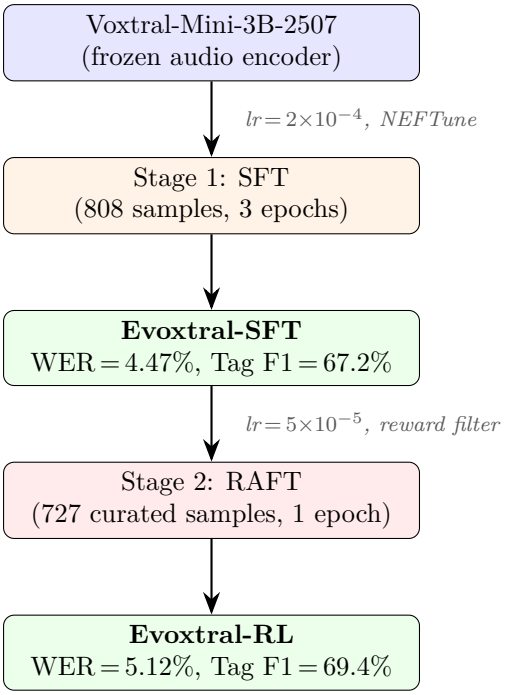

Figure 2: Full two-stage Evoxtral training pipeline from base Voxtral-Mini-3B to the two released model variants.

Table 2: Core evaluation results on Evoxtral-Bench (50 samples). Bold indicates best per metric. ↓ lower is better; ↑ higher is better.

Metric	Base Voxtral	Evoxtral SFT	Evoxtral RL	Best
WER ↓	6.64%	**4.47%**	5.12%	SFT
CER ↓	2.72%	**1.23%**	1.48%	SFT
Tag F1 ↑	22.0%	67.2%	**69.4%**	RL
Tag Precision ↑	22.0%	67.4%	**68.5%**	RL
Tag Recall ↑	22.0%	69.4%	**72.7%**	RL
Emphasis F1 ↑	42.0%	84.0%	**86.0%**	RL
Tag Hallucination ↓	0.0%	**19.3%**	20.2%	SFT

- **CER** (Character Error Rate, ↓) — character-level accuracy via `jiwer`
- **Tag F1** (↑) — token-level F1 on predicted vs. reference tag multisets
- **Tag Precision** (↑) — fraction of predicted tags present in reference
- **Tag Recall** (↑) — fraction of reference tags captured by the model
- **Tag Hallucination Rate** (↓) — fraction of predicted tags absent from reference
- **Emphasis F1** (↑) — F1 on CAPITALIZED emphasis words

Per-tag F1 is additionally computed across all 17 tag types.

4.2 Core Results

The base Voxtral model achieves 22.0% Tag F1, suggesting some limited native capability to produce expressive tokens but with low precision and recall. SFT provides the dominant improvement: WER decreases by 33% relative (6.64% → 4.47%) and Tag F1 increases by

Table 3: Per-tag F1 scores for SFT and RL on Evoxtral-Bench. Support indicates the number of test samples containing each tag.

Tag	SFT F1	RL F1	Δ	Support
[sighs]	1.000	1.000	—	9
[clears throat]	0.889	**1.000**	+12.5%	8
[gasps]	0.957	0.957	—	12
[pause]	0.885	**0.902**	+1.9%	25
[nervous]	0.800	**0.846**	+5.8%	13
[stammers]	**0.889**	0.842	−5.3%	8
[laughs]	0.800	**0.815**	+1.9%	12
[sad]	0.667	**0.750**	+12.4%	4
[whispers]	0.636	**0.667**	+4.9%	13
[crying]	**0.750**	0.571	−23.9%	5
[excited]	**0.615**	0.571	−7.2%	5
[shouts]	0.400	**0.500**	+25.0%	3
[calm]	0.200	**0.400**	+100%	6
[frustrated]	0.444	0.444	—	3
[angry]	0.667	0.667	—	2
[confused]	0.000	0.000	—	1
[scared]	0.000	0.000	—	1

45 percentage points (22.0% → 67.2%). RAFT further refines tag metrics: Tag F1 improves by 2.2 pp (67.2% → 69.4%), Tag Recall by 3.3 pp (69.4% → 72.7%), and Emphasis F1 by 2.0 pp (84.0% → 86.0%). However, RAFT introduces a small WER regression (4.47% → 5.12%), reflecting a Pareto tradeoff between transcription accuracy and expressive richness.

Tag hallucination—predicted tags absent from the reference—is 19.3% for SFT and 20.2% for RL. The base model has 0% hallucination because it rarely predicts any tags at all.

4.3 Per-Tag F1 Breakdown

RAFT improves 9 tags, maintains 4 stable, and regresses on 3 (Table 3). The largest gains are observed for [calm] (+100%, 0.200 → 0.400), [shouts] (+25.0%), [clears throat] (+12.5%), and [sad] (+12.4%). Regressions are noted for [crying] (−23.9%), [excited] (−7.2%), and [stammers] (−5.3%). The two zero-F1 tags ([confused], [scared]) each appear only once in the test set, making estimation unreliable.

5 Analysis and Discussion

SFT as the primary driver of improvement. The SFT stage accounts for the vast majority of the performance gain: WER drops 33% relative and Tag F1 increases by 45 percentage points. This aligns with findings from GRPO-based ASR work Shi et al. [2025], [2025a,b] suggesting that a well-supervised initial adaptation is a strong foundation for subsequent RL refinement.

The WER–Tag tradeoff. RAFT improves tag metrics at the cost of a modest WER regression (4.47% → 5.12%). This suggests the existence of a Pareto frontier between transcription accuracy and expressive richness: optimizing for tag generation pushes the model toward producing more tags, which can introduce minor word-level errors. This motivates releasing two model variants—Evoxtral-SFT for accuracy-critical applications (e.g., professional transcription) and

Evoxtral-RL for expressiveness-critical applications (e.g., downstream TTS synthesis with ElevenLabs v3 ElevenLabs [2025]).

Tag hallucination. Approximately 20% of predicted tags are not present in the reference transcription. Hallucination may occur when the model infers an expressive tone from acoustic cues that are present in the audio but absent or differently annotated in the reference. This may partly reflect annotation noise in synthetic data rather than pure model error. Future work should address this with contrastive or calibration-based training objectives.

Effect of NEFTune. Applying NEFTune Jain et al. [2024] with noise alpha = 5.0 during SFT provided a regularization benefit on the small 808-sample training set, consistent with Jain et al.'s findings on instruction-following tasks. Ablating this component was not feasible within hackathon time constraints but remains a planned analysis.

Rare tag performance. Tags with very low test support (`[confused]`, `[scared]`, support = 1) have zero F1, which is uninformative. Tags with support 2–6 show high variance in F1 estimates. A larger, more balanced evaluation set would provide more reliable per-tag metrics.

Reward function design. The RAFT reward (Equation 1) explicitly encodes the design preference for equal weight on accuracy and expressiveness. The 0.2 weight on hallucination acts as a weak regularizer. An ablation across reward weightings would quantify the sensitivity of the final model to this design choice.

6 Limitations

- **Synthetic training data.** All 1,010 samples are synthesized using ElevenLabs TTS v3 ElevenLabs [2025]. The acoustic properties of synthetic speech differ from natural human speech. Performance on natural speech recordings may differ.
- **Tag hallucination.** Approximately 20% of predicted tags in the RL model are not present in the reference, which may limit applicability in settings requiring precise expressive annotation.
- **Rare tag coverage.** Seventeen tag types are represented, but several occur in fewer than 5 test samples. Per-tag F1 estimates for rare categories are unreliable.
- **English only.** The dataset and training are English-only. Generalization to other languages is not evaluated.
- **Small dataset.** 808 training samples is a small fine-tuning set. Scaling to thousands of examples with natural speech could substantially improve performance.
- **Evaluation scope.** Evoxtral-Bench covers 50 test samples. A larger evaluation set would yield more statistically robust estimates.

7 Conclusion

We presented Evoxtral, a LoRA-adapted version of Voxtral-Mini-3B-2507 Mistral AI [2025] that produces expressive tagged transcriptions using ElevenLabs v3 audio tags ElevenLabs [2025]. Our two-stage training pipeline—SFT followed by RAFT Yuan et al. [2023]—demonstrates that expressive tagging capability can be effectively injected into a pre-trained ASR model with parameter-efficient fine-tuning Hu et al. [2022], Mangrulkar et al. [2022].

SFT achieves a 33% relative WER reduction and a 45 percentage-point improvement in Tag F1 over the base model. RAFT further improves tag recall and F1 by targeting tag generation

quality directly through a rule-based reward signal, at a modest transcription accuracy cost. The two resulting model variants cover a Pareto frontier between accuracy and expressiveness, allowing practitioners to select the appropriate trade-off for their application.

Future directions include: (1) collecting natural speech data with crowd-sourced expressive annotations to reduce the synthetic data gap; (2) replacing RAFT with GRPO Shi et al. [2025] or DPO Mistral AI [2025] for more sample-efficient RL training; (3) expanding to multilingual settings leveraging Voxtral's multilingual audio encoder; and (4) developing joint ASR+TTS evaluation protocols that measure downstream TTS quality when Evoxtral output is used as input to ElevenLabs v3 ElevenLabs [2025].

References

Mistral AI. Voxtral. *arXiv preprint arXiv:2507.13264*, 2025. https://arxiv.org/abs/2507.13264

Shi, B. et al. Group relative policy optimization for speech recognition. *arXiv preprint arXiv:2509.01939*, 2025. https://arxiv.org/abs/2509.01939

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *Proceedings of ICLR*, 2022. https://arxiv.org/abs/2106.09685

Yuan, Z., Yuan, H., Li, C., Dong, G., Tan, C., and Zhou, C. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023. https://arxiv.org/abs/2308.01825

Jain, N., Chiang, P., Yeh, Y., Kirchenbauer, J., et al. NEFTune: Noisy embeddings improve instruction finetuning. In *Proceedings of ICLR*, 2024. https://arxiv.org/abs/2310.05914

ElevenLabs. Text-to-speech v3 audio tags. ElevenLabs Developer Documentation, 2025. https://elevenlabs.io/docs/api-reference/text-to-speech

JiWER: Evaluate your speech recognition system. Python library for ASR evaluation metrics. https://github.com/jitsi/jiwer

Advancing speech understanding in speech-aware language models with GRPO. *arXiv preprint arXiv:2509.16990*, 2025. https://arxiv.org/abs/2509.16990

Explore the reinforcement learning for LLM-based ASR and TTS system. *arXiv preprint arXiv:2509.18569*, 2025. https://arxiv.org/abs/2509.18569

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. PEFT: State-of-the-art parameter-efficient fine-tuning methods. HuggingFace, 2022. https://github.com/huggingface/peft

Morris, A.C., Maier, V., and Green, P. From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition. In *Proceedings of INTERSPEECH*, 2004.

Reinforcement learning outperforms supervised fine-tuning: A case study on audio question answering. *arXiv preprint arXiv:2503.11197*, 2025. https://arxiv.org/abs/2503.11197