
Multimodal Data and Resource Efficient Device-directed Speech Detection with Large Foundation Models

Dominik Wagner^{1*}, Alexander Churchill², Siddharth Sigtia², Panayiotis Georgiou²,
Matt Mirsamadi², Aarshee Mishra², Erik Marchi²

¹TH Nürnberg, ²Apple

dominik.wagner@th-nuernberg.de, alex.churchill@apple.com,
sidsigtia@apple.com, panayiotis_georgiou@apple.com,
smirsamadi@apple.com, aarshee_mishra@apple.com, emarchi@apple.com

Abstract

Interactions with virtual assistants typically start with a trigger phrase followed by a command. In this work, we explore the possibility of making these interactions more natural by eliminating the need for a trigger phrase. Our goal is to determine whether a user addressed the virtual assistant based on signals obtained from the streaming audio recorded by the device’s microphone. We address this task by combining 1-best hypotheses and decoder signals from an automatic speech recognition system with acoustic representations from an audio encoder as input features to a large language model (LLM). In particular, we are interested in data and resource efficient systems that require only a small amount of training data and can operate in scenarios with only a single frozen LLM available on a device. For this reason, our model is trained on 80k or less examples of multimodal data using a combination of low-rank adaptation and prefix tuning. We compare the proposed system to unimodal baselines and show that the multimodal approach achieves lower equal-error-rates (EERs), while using only a fraction of the training data. We also show that low-dimensional specialized audio representations lead to lower EERs than high-dimensional general audio representations.

1 Introduction

Speech-based virtual assistants allow users to interact with devices such as phones, watches, and loudspeakers via voice commands. To distinguish audio that is directed towards a device from background speech, a trigger phrase or the press of a button usually precedes the user command [1]. The problem of detecting a trigger phrase is referred to as, wake-word detection [2, 3], voice trigger detection [4, 5], or keyword spotting [6, 7, 8, 9]. To create a more natural conversation flow, subsequent commands after the initial interaction should not require the trigger phrase. Device-directed speech detection is concerned with determining whether a virtual assistant was addressed or not, without a trigger cue preceding the voice command at all times [10, 11, 12]. Device-directed speech detection systems are exposed to information from all kinds of in-domain (voice commands) and out-of-domain (e.g. background speech, ambient sounds, appliances etc.) signals. Previous works use a combination of acoustic and lexical features to encode the relevant information in those signals [10, 11, 13, 14, 15].

Recent studies have extended LLMs with the ability to process non-lexical input modalities, such as audio and video data [16, 17, 18, 19, 20, 21]. Inspired by these efforts, we explore a LLM-based multimodal model to differentiate between directed and non-directed audio in interactions with a virtual assistant. Our goal is to determine whether the user addressed the assistant using signals obtained from the streaming audio captured by the device’s microphone.

*Work done during an internship at Apple.

The proposed model uses acoustic features obtained from a pretrained audio encoder in combination with decoder signals, such as acoustic cost, as well as 1-best hypotheses from an ASR system. The acoustic features and decoder signals are represented as learnable fixed-length prefixes, which are concatenated with the token embeddings of the 1-best hypotheses (cf. Figure 1). The system is optimized to generate decisions about device-directedness by jointly learning from all modalities using a combination of prefix tuning [22] and low-rank adaptation (LoRA) [23].

We analyze this task in a scenario in which (1) only a limited amount of training data is available and (2) only a pretrained LLM with frozen weights is usable on a resource-constrained device (e.g. a smartphone). Furthermore, we compare the effectiveness of high-dimensional representations obtained from a large generic audio foundation model with lower-dimensional representations from a small audio encoder trained on in-domain data.

2 Feature Extraction

1-best Hypotheses and ASR Decoder Signals The text part of the data was transcribed with an on-device joint CTC-attention based end-to-end speech recognition system [24] trained on in-domain data, comparable to the one used in [25]. Inspired by [10, 11], we extract 4 additional utterance-level signals that are generated by a decoder based on weighted finite-state transducers [26]. For the most likely hypothesis in the N-best set of hypotheses, we extract the average of the graph cost associated with each word in the hypothesis, the average of the acoustic cost, and the average of the word-level posterior confidence scores. The graph cost is the sum of language model cost, transition probabilities, and pronunciation cost [27]. The acoustic cost is the negative log-likelihood of the tokens from the decoder. Additionally, we include the average number of alternative word options for each word in the 1-best hypothesis. Finally, we scale the feature values along each signal dimension into the unit interval $[0, 1]$ across the dataset.

Audio Representations We compare two pretrained models as backbones to extract audio representations. The first model is the medium version of Whisper (769M parameters) [28]. Whisper is expected to generalize well across domains and languages, since it was trained on 680k hours of speech data and is therefore well-suited for our task. We extract 1024-dimensional representations at the last encoder layer of Whisper. Additionally, we explore a specialized and lightweight on-device model for acoustic feature extraction. We choose the Unified Acoustic Detector (UAD) for false trigger mitigation described in [29] as an alternative feature extractor. The model is trained to detect unintended invocations of devices such as smartphones. It has ≈ 6 million parameters and consists of a shared transformer-based encoder [30], followed by task-specific classification heads. We extract 256-dimensional representations at one of the task-specific classification heads.

3 Method

Our system consists of three main components (cf. Figure 1). The first main component is a frozen audio encoder (either Whisper or UAD), which extracts a sequence of latent representations in \mathbb{R}^N from the input audio. The second main component comprises two feedforward mapping networks, M_1 and M_2 , which translate the extracted audio features and the utterance-level ASR decoder signals into the latent space of the token embeddings. The third main component is a decoder-only LLM that generates text given the prefix representations obtained via M_1 and M_2 as well as the 1-best ASR hypotheses, and a text prompt (i.e., “directed decision:” in Figure 1).

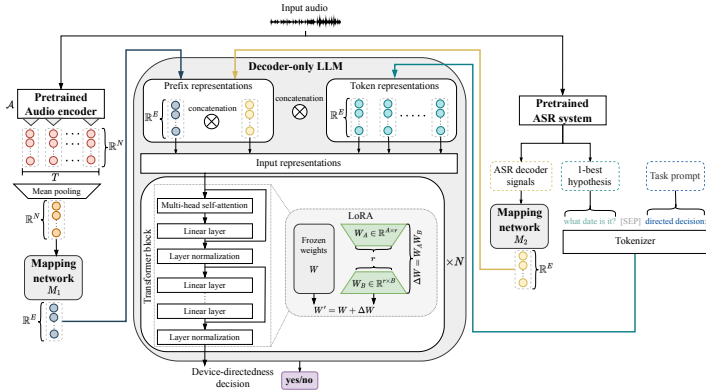


Figure 1: Architecture of the multimodal system. The weights of the LoRA modules are trained along with the weights of M_1 and M_2 . All other components remain frozen.

The task is to generate a decision on whether an unseen utterance is directed towards a device or not. The model is trained on multimodal data that contains L examples of audio, ASR decoder, and

1-best hypotheses $\{(x^i, d^i, t^i)\}_{i=1}^L$. The 1-best hypotheses are represented by a sequence of tokens $t^i = (t_1^i, \dots, t_l^i)$, which are padded to a maximum length l . The input waveform x^i is transformed into log-magnitude Mel spectrogram features via the transformation \mathcal{F} . The spectrogram feature input $\mathcal{F}(x^i)$ is then processed to obtain a sequence of embeddings in \mathbb{R}^N of length T using the audio encoder \mathcal{A} (either Whisper or UAD). Mean pooling is applied to these representations along the time dimension to generate a single vector in \mathbb{R}^N per utterance. M_1 is then used to map the aggregated embedding to a vector in the prefix token space of the LLM:

$$a^i = M_1 \left(\frac{1}{T} \sum_{t=1}^T h_t \right) \in \mathbb{R}^{1 \times E}, \quad \mathcal{A}(\mathcal{F}(x^i)) = [h_1 \dots h_T]^\top \in \mathbb{R}^{T \times N}. \quad (1)$$

The resulting representation a^i has the same dimensionality E as the token embeddings. M_2 is used to generate a latent prefix b^i in \mathbb{R}^E for the decoder signal features d^i . The audio prefix a^i and the decoder signal prefix b^i are then concatenated to the token embeddings of the corresponding 1-best hypothesis t^i , and the concatenated input features are presented to the LLM. The training objective is to predict directedness tokens conditioned on the prefix, the decoder signals, and the 1-best hypothesis tokens in an autoregressive fashion. We utilize cross entropy loss to train the parameters θ of the model:

$$\mathcal{L}_\theta = - \sum_{i=1}^L \sum_{j=1}^l \log p_\theta(t_j^i | a^i, b^i, t_1^i, \dots, t_{j-1}^i). \quad (2)$$

During inference, the device-directedness decision is made based on the score $p_\theta(Y = \text{yes} | c)$, where Y is a discrete random variable that can take one of m tokens y_1, \dots, y_m from the vocabulary \mathcal{V} , and $p_\theta(Y = \text{yes} | c) + p_\theta(Y = \text{no} | c) \simeq 1$. The context c is determined by the multimodal features, i.e., $c = (a^i, b^i, t_1^i, \dots, t_{j-1}^i)$.

Large Language Models We focus on decoder-only LLMs, since this architecture choice has demonstrated stronger capabilities [31] than encoder-only and encoder-decoder systems, such as BERT [32] and T5 [33], on a wide range of tasks. We compare the 7B parameter versions of Falcon [34] and RedPajama [35] in our experiments.

Mapping Networks The mapping networks M_1 and M_2 translate between the latent space of the audio encoder and the lexical embedding space of the LLM. All audio features and ASR decoder signals are transformed into \mathbb{R}^E sized prefixes, where E is the latent dimension of the LLM. Both mapping networks share the same architecture, consisting of one hidden linear layer with $\frac{E}{2}$ units and hyperbolic tangent activation. The models are trained with a dropout [36] probability of 10%.

Low-rank Adaptation We employ low-rank adaptation (LoRA) [23] to finetune the LLM without directly changing its weights. In the LoRA method, weights of dense layers in large pretrained models are summed with linear low-rank adapter modules. These adapter modules are small trainable matrices, which are included into the architecture and optimized on behalf of the underlying LLM weights. We attach adapter modules to the query q and value v projection matrices, as well as the dense layers d of each transformer block. We employ the configuration $r_q = r_v = d = 8$, $\alpha = 32$ and train the adapters with a dropout probability of 10%. The parameter r is the rank of the adaptation matrices, and α is a scaling factor to adjust the magnitude of the adaptation. The LoRA approach allows us to use less training data [37] and enables the reuse of a generic LLM deployed on a device.

Unimodal Baselines Unimodal versions of our framework are trained by providing text, decoder signals, or audio representations as the only input source to the LLM. In the *text-only* variant, the mapping networks M_1 and M_2 are removed, and the only input features are the 1-best hypotheses of the ASR system (cf. Figure 1). In the *audio-only* variant, the decoder signals including M_2 and the 1-best hypotheses are removed from the system. The *decoder-signal-only* system relies only on the decoder signal input, which is transformed via M_2 . Hence, M_1 and the 1-best hypotheses are removed from the overall system.

4 Experiments

Data The full training data is a balanced set of ≈ 40 k directed utterances and ≈ 40 k non-directed utterances, similar to the set used in [29] and [38]. The evaluation data is a combined set of two in-house corpora with ≈ 14 k device-directed utterances and ≈ 23 k non-directed utterances. The total duration of the evaluation data is ≈ 35 hours. Approximately 29% of the device-directed training examples start with a trigger phrase and $\approx 12\%$ of the device-directed evaluation utterances start with a trigger phrase. The remaining device-directed utterances are triggerless interactions with a virtual assistant. All utterances in the training and evaluation data are randomized and anonymized. The dataset statistics are summarized in Table 2 of Appendix B.

Table 1: Comparison EERs on the evaluation set. “Uni” refers to unimodal experiments, and “Multi” refers to multimodal experiments. “Modality” indicates the modalities used in the experiment (t = text, a = audio, b = decoder signals). “Train Size” shows the number of training examples used in the experiment. “# Param” is the number of trainable parameters. We report the sum of the parameters of the mapping networks and LoRA.

Experiment	LoRA	Modality	Train Size	Falcon 7B			RedPajama 7B		
				# Param	EER Whisper	EER UAD	# Param	EER Whisper	EER UAD
Uni 1	✓	t	80k	16M	12.97%	12.97%	17M	12.90%	12.90%
Uni 2	✓	a	80k	29M	10.45%	9.31%	27M	10.78%	8.99%
Uni 3	✓	b	80k	26M	36.90%	36.90%	25M	35.04%	35.04%
Multi 1	✓	t, b	80k	26M	13.39%	13.39%	25M	12.86%	12.96%
Multi 2	✓	a, b	80k	39M	14.94%	9.92%	35M	14.80%	10.71%
Multi 3	✓	t, a	80k	29M	9.96%	8.76%	27M	9.89%	8.44%
Multi 4	✓	t, a, b	80k	39M	8.80%	8.23%	35M	9.45%	8.52%
Multi 5 (frozen LLM)	✗	t, a, b	80k	23M	10.52%	11.49%	18M	10.90%	12.26%
Multi 4.1	✓	t, a, b	40k	39M	10.19%	8.38%	35M	10.20%	8.47%
Multi 4.2	✓	t, a, b	20k	39M	10.67%	9.05%	35M	10.91%	8.84%
Multi 4.3	✓	t, a, b	10k	39M	11.71%	8.84%	35M	11.66%	9.69%
Multi 4.4	✓	t, a, b	5k	39M	12.76%	9.77%	35M	12.11%	9.65%
Multi 4.5	✓	t, a, b	1k	39M	15.39%	12.56%	35M	17.09%	11.87%

Results and Discussion The equal-error-rates (EERs) for our experiments are summarized in Table 1. The unimodal baselines (Uni 1-3) are the *text-only* (t), *audio-only* (a), and *decoder-signal-only* (b) versions of the proposed system. Using only the audio modality (Uni 2) yields lower EERs than using only the text modality (Uni 1), irrespective of the underlying LLM and audio encoder. Furthermore, using the specialized UAD representations leads to lower EERs than using Whisper representations in experiment Uni 2. Decoder signals (Uni 3) provide the weakest overall signal ($EER = 36.90\%$ with Falcon and $EER = 35.04\%$ with RedPajama). The best system configuration (Multi 4) uses all 80k available training examples and combines information from text, audio, as well as decoder signals. Multi 4 with Falcon shows an EER of 8.80% using Whisper as the audio encoder and an EER of 8.23% with the UAD backbone, which translates to relative improvements of $\approx 16\%$ and $\approx 12\%$ over the corresponding *audio-only* models (Uni 2). In Multi 5, only M_1 and M_2 are trained (i.e., the underlying LLM is frozen and no LoRA modules are attached). This configuration shows worse results than Multi 4, indicating that training the mapping networks alone is not sufficient to achieve low EERs. The experiments Multi 4.1 to Multi 4.5 are the same as Multi 4 but with a stepwise reduction of the training data (from 40k examples to 1k examples). The multimodal system with Falcon and the UAD backbone trained on only 10k examples (Multi 4.3) still performs better than the *audio-only* model trained on 80k examples (EERs of 8.84% and 9.31%). This is not the case when Whisper representations are used instead (EERs of 11.71% and 10.45%). Additional experiments showing the impact of using the smallest (39M) and largest (1.5B) versions of Whisper as audio feature extractors can be found in Table 4 of Appendix D.

In contrast to other systems for device-directed detection [11, 13, 39], our approach requires only a small amount of training data. We see that audio representations from a pretrained Whisper model perform well in this low data environment. However, the model can be further improved by replacing these unspecialized representations with specialized ones obtained from the smaller UAD model. This effect is amplified in very low data environments (see Multi 4.1-4.5). While we observe a strong EER increase with less training data (e.g. from 8.80% with 80k examples to 15.39% with 1k examples using Falcon) when Whisper representations are used, the EER increase is less pronounced with UAD representations (e.g. from 8.23% with 80k examples to 12.56% with 1k examples using Falcon). We hypothesize that in low data environments the model relies more on what it already knows (i.e., the acoustic information encoded in the in-domain UAD model) and the amount of training data is not sufficient to learn how the acoustic information encoded in unspecialized representations can be utilized accordingly.

5 Conclusions

In this work, we described a multimodal model to distinguish device-directed utterances from background speech. Our approach made use of knowledge encoded in pretrained foundation models and effectively combined decoder signals with audio and lexical information. The system can be trained on small amounts of data and operates in scenarios, where only a single frozen LLM is available on a resource-constrained device. We achieved lower EERs than unimodal baselines, while using only a fraction of the training data. Furthermore, low-dimensional audio representations from a small specialized feature encoder outperformed high-dimensional general representations from a larger audio foundation model and showed more stable results in environments with very low data availability (i.e., $< 80k$ utterances).

References

- [1] Siri Team, “Voice trigger system for Siri.” <https://machinelearning.apple.com/research/voice-trigger>, 2023.
- [2] C. Jose, Y. Mishchenko, T. Sénéchal, A. Shah, A. Escott, and S. N. P. Vitaladevuni, “Accurate Detection of Wake Word Start and End Using a CNN,” in *Interspeech*, 2020.
- [3] A. Ghosh, M. Fuhs, D. Bagchi, B. Farahani, and M. Woszczyna, “Low-resource Low-footprint Wake-word Detection using Knowledge Distillation,” in *Interspeech*, 2022.
- [4] S. Sigtia, R. Haynes, H. Richards, E. Marchi, and J. Bridle, “Efficient Voice Trigger Detection for Low Resource Hardware,” in *Interspeech*, 2018.
- [5] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, and J. Bridle, “Multi-task learning for speaker verification and voice trigger detection,” in *ICASSP*, 2020.
- [6] T. Sainath and C. Parada, “Convolutional neural networks for small-footprint keyword spotting,” in *Interspeech*, 2015.
- [7] A. H. Michaely, X. Zhang, G. Simko, C. Parada, and P. Aleksic, “Keyword spotting for google assistant using contextual speech recognition,” in *ASRU*, 2017.
- [8] S. Cornell, T. Balestri, and T. Sénéchal, “Implicit acoustic echo cancellation for keyword spotting and device-directed speech detection,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023.
- [9] D. Ng, R. Zhang, J. Q. Yip, C. Zhang, Y. Ma, T. H. Nguyen, C. Ni, E. S. Chng, and B. Ma, “Contrastive speech mixup for low-resource keyword spotting,” in *ICASSP*, 2023.
- [10] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and L. Heck, “Learning when to listen: detecting system-addressed speech in human-human-computer dialog,” in *Interspeech*, 2012.
- [11] S. H. Mallidi, R. Maas, K. Goehner, A. Rastrow, S. Matsoukas, and B. Hoffmeister, “Device-directed Utterance Detection,” in *Interspeech*, 2018.
- [12] V. Garg, O. Rudovic, P. Dighe, A. H. Abdelaziz, E. Marchi, S. Adya, C. Dhir, and A. Tewfik, “Device-Directed Speech Detection: Regularization via Distillation for Weakly-Supervised Models,” in *Interspeech*, 2022.
- [13] K. Gillespie, I. C. Konstantakopoulos, X. Guo, V. T. Vasudevan, and A. Sethy, “Improving device directedness classification of utterances with semantic lexical features,” in *ICASSP*, 2020.
- [14] H. Sato, Y. Shinohara, and A. Ogawa, “Multi-modal modeling for device-directed speech detection using acoustic and linguistic cues,” *Acoustical Science and Technology*, vol. 44, no. 1, pp. 40–43, 2023.
- [15] D. Bekal, S. Srinivasan, S. Ronanki, S. Bodapati, and K. Kirchhoff, “Contextual Acoustic Barge-In Classification for Spoken Dialog Systems,” in *Interspeech*, 2022.
- [16] R. Mokady, A. Hertz, and A. H. Bermanto, “ClipCap: CLIP prefix for image captioning,” 2021. arXiv:2111.09734.
- [17] D. Driess *et al.*, “PaLM-E: An embodied multimodal language model,” 2023. arXiv:2303.03378.
- [18] Y. Fathullah *et al.*, “Prompting large language models with speech recognition abilities,” 2023. arXiv:2307.11795.
- [19] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. Glass, “Listen, think, and understand,” 2023. arXiv:2305.10790.
- [20] M. Kim, K. Sung-Bin, and T.-H. Oh, “Prefix tuning for automated audio captioning,” in *ICASSP*, 2023.
- [21] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, “Pengi: An audio language model for audio tasks,” 2023. arXiv:2305.11834.
- [22] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Online), pp. 4582–4597, Association for Computational Linguistics, Aug. 2021.
- [23] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *ICLR*, 2022.
- [24] S. Kim, T. Hori, and S. Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” in *ICASSP*, 2017.
- [25] M. Bleeker, P. Swietojanski, S. Braun, and X. Zhuang, “Approximate Nearest Neighbour Phrase Mining for Contextual Speech Recognition,” in *Interspeech*, 2023.
- [26] Y. Miao, M. Gowayyed, and F. Metze, “EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” in *ASRU*, 2015.

- [27] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian, K. Riedhammer, K. Veselý, and N. T. Vu, “Generating exact lattices in the WFST framework,” in *ICASSP*, 2012.
- [28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” 2022.
- [29] O. Rudovic, W. Chang, V. Garg, P. Dighe, P. Simha, J. Berkowitz, A. H. Abdelaziz, S. Kajarekar, E. Marchi, and S. Adya, “Less is more: A unified architecture for device-directed speech detection with multiple invocation types,” in *ICASSP*, 2023.
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.
- [31] T. B. Brown *et al.*, “Language models are few-shot learners,” 2020. arXiv:2005.14165.
- [32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.
- [33] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *JMLR*, vol. 21, no. 140, pp. 1–67, 2020.
- [34] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo, “Falcon-40B: an open large language model with state-of-the-art performance,” 2023.
- [35] Together Computer, “RedPajama: An open source recipe to reproduce LLaMA training dataset,” 2023.
- [36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [37] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” 2020.
- [38] P. Dighe, P. Nayak, O. Rudovic, E. Marchi, X. Niu, and A. Tewfik, “Audio-to-intent using acoustic-textual subword representations from end-to-end asr,” in *ICASSP*, 2023.
- [39] O. Rudovic, A. Bindal, V. Garg, P. Simha, P. Dighe, and S. Kajarekar, “Streaming on-device detection of device directed speech from voice and touch-based invocation,” in *ICASSP*, 2022.

Appendix

A Acknowledgements

We would like to express our sincere gratitude to John Bridle, Pranay Dighe, Sachin Kajarekar, Oggi Rudovic, Ahmed Tewfik and Barry Theobald for their support and their comprehensive feedback on this work. We also thank Seanie Lee for the numerous helpful discussions.

B Datasets

Table 2: Summary of the data used in our experiments.

Label	Examples (#)		Total Duration (hours)		Duration per Utterance (seconds)		Words per Utterance (#)	
	Train	Eval	Train	Eval	Train	Eval	Train	Eval
Directed	40568	14396	58.88	12.05	5.22±6.97	3.01±1.89	5.34±3.26	5.50±3.64
Non-directed	40062	22958	67.31	23.37	6.04±5.33	3.66±3.67	6.48±9.29	9.42±17.28
Combined	80630	37354	126.19	35.42	5.63±6.22	3.41±3.12	5.90±6.97	7.89±13.81

C Examples

Table 3: Examples of 1-best hypotheses of device-directed and non-directed utterances.

Directed	“Set an alarm for 8 AM”
Directed	“Tell me a joke”
Directed	“What’s the temperature”
Non-directed	“Excellent thank you very much”
Non-directed	“Can we talk”
Non-directed	“I was trying to do it”

D Additional Experiments

Table 4: Impact of changing the size of the Whisper audio foundation model. “Whisper Tiny” has ≈ 39 M parameters and the audio representation dimension is \mathbb{R}^{384} . “Whisper Large” has ≈ 1.5 B parameters and the audio representation dimension is \mathbb{R}^{1280} . The LoRA configuration is the same as in Table 1 ($r_q = r_v = d = 8, \alpha = 32$).

Experiment	Modality	Train Size	Falcon 7B		RedPajama 7B	
			EER Whisper Tiny	EER Whisper Large	EER Whisper Tiny	EER Whisper Large
Uni 2	a	80k	14.66%	10.14%	13.84%	9.76%
Multi 4	t, a, b	80k	10.56%	10.03%	9.75%	9.02%

Table 5: Alternative LoRA configuration for Falcon 7B and RedPajama 7B. LoRA modules are only attached to q and v and $r_q = r_v = 64, \alpha = 16$ is used.

Experiment	Modality	Train Size	Falcon 7B			RedPajama 7B		
			# Param	EER Whisper	EER UAD	# Param	EER Whisper	EER UAD
Uni 1	t	80k	19M	12.59%	12.59%	33M	12.88%	12.88%
Uni 2	a	80k	32M	10.53%	12.52%	43M	10.92%	9.16%
Multi 3	t, a	80k	32M	9.33%	9.36%	43M	9.44%	8.45%
Multi 4	t, a, b	80k	42M	10.13%	10.55%	51M	9.72%	9.75%
Multi 4.1	t, a, b	40k	42M	9.97%	11.10%	51M	9.86%	9.19%
Multi 4.2	t, a, b	20k	42M	11.07%	11.55%	51M	10.63%	9.73%
Multi 4.3	t, a, b	10k	42M	10.91%	12.96%	51M	12.00%	9.95%
Multi 4.3	t, a, b	5k	42M	12.53%	12.49%	51M	12.08%	10.81%
Multi 4.5	t, a, b	1k	42M	17.63%	13.36%	51M	14.71%	14.17%

Table 6: Alternative LoRA configurations for Falcon 7B. Note that the configuration in the middle column ($r_q = r_v = d = 8, \alpha = 32$) is the same as in Table 1.

Experiment	Modality	Train Size	$r_q = r_v = 8, \alpha = 32$		$r_q = r_v = d = 8, \alpha = 32$		$r_q = r_v = d = 64, \alpha = 16$	
			EER Whisper	EER UAD	EER Whisper	EER UAD	EER Whisper	EER UAD
Uni 1	t	80k	12.53%	12.53%	12.97%	12.97%	12.47%	12.47%
Uni 2	a	80k	10.33%	13.81%	10.45%	9.31%	10.95%	9.25%
Uni 3	b	80k	34.24%	34.24%	36.90%	36.90%	35.42%	35.42%
Multi 1	t, b	80k	13.19%	13.19%	13.39%	13.39%	12.99%	12.99%
Multi 2	a, b	80k	16.85%	10.49%	14.94%	9.92%	13.48%	10.35%
Multi 3	t, a	80k	9.09%	9.08%	9.96%	8.76%	9.51%	8.00%
Multi 4	t, a, b	80k	9.17%	10.92%	8.80%	8.23%	9.09%	8.00%

Table 7: Alternative LoRA configuration for RedPajama 7B. Note that the configuration in the middle column ($r_q = r_v = d = 8, \alpha = 32$) is the same as in Table 1.

Experiment	Modality	Train Size	$r_q = r_v = 8, \alpha = 32$		$r_q = r_v = d = 8, \alpha = 32$		$r_q = r_v = d = 64, \alpha = 16$	
			EER Whisper	EER UAD	EER Whisper	EER UAD	EER Whisper	EER UAD
Uni 1	t	80k	13.15%	13.15%	12.90%	12.90%	12.87%	12.87%
Uni 2	a	80k	10.76%	8.93%	10.78%	8.99%	11.40%	8.82%
Uni 3	b	80k	33.66%	33.66%	35.04%	35.04%	35.70%	35.70%
Multi 1	t, b	80k	13.00%	13.00%	12.86%	12.96%	13.08%	13.26%
Multi 2	a, b	80k	13.12%	10.00%	14.80%	10.71%	13.90%	10.72%
Multi 3	t, a	80k	9.84%	9.70%	9.89%	8.44%	9.57%	8.27%
Multi 4	t, a, b	80k	9.43%	9.76%	9.45%	8.52%	9.37%	8.55%