Device-Directed Speech Detection for Follow-up Conversations Using Large Language Models

Ognjen (Oggi) Rudovic*, Pranay Dighe*, Yi Su*, Vineet Garg, Sameer Dharur, Xiaochuan Niu, Ahmed H. Abdelaziz, Saurabh Adya, Ahmed Tewfik

Apple

Abstract

Follow-up conversations with virtual assistants (VAs) enable a user to seamlessly interact with a VA without the need to repeatedly invoke it using a keyword (after the first query). Therefore, accurate Device-directed Speech Detection (DDSD) from the follow-up queries is critical for enabling naturalistic user experience. To this end, we explore the notion of Large Language Models (LLMs) and model the first query when making inference about the follow-ups (based on the ASR-decoded text), via prompting of a pretrained LLM, or by adapting a binary classifier on top of the LLM. In doing so, we also exploit the ASR uncertainty when designing the LLM prompts. We show on the real-world dataset of follow-up conversations that this approach yields large gains (20-40% reduction in false alarms at 10% fixed false rejects) due to the joint modeling of the previous speech context and ASR uncertainty, compared to when follow-ups are modeled alone.

1 Introduction

Virtual assistants (VAs) are at the core of smart devices (e.g., mobile phones, smart speakers, wearables, etc.) as they aim to enable a naturalistic voice-based interaction between a user and a device. For VAs to respond to the user requests reliably, they need to infer whether the user is talking to the device or not. For instance, the user could be talking to someone else, and/or there could be side-speech conversations, background noise, etc. Therefore, classifying accurately if the user's speech is device-directed is critical for providing relevant responses, and to avoid interfering with the user's interactions which are not device-directed, i.e., intended for the VA. This task is often referred to as the device-directed speech detection (DDSD) [1; 2; 3; 4]. Most existing works on DDSD focus on detection from single queries of the user, often beginning with a wakeword (e.g., "Hey Google", "Hey Siri", "Alexa", and so on). Such isolated utterances are usually a *complete* question or a task request from a user to the VA, and often do not require additional context to determine if the speech is the VA directed. In this work, we address the task of DDSD in follow-up conversations, where the user's first query starts with a wakeword (that is easier to detect with high accuracy by existing systems for the wakeword detection [5]), potentially followed by another query (termed as the "follow-up"), as a continuation of the conversation with the VA. The follow-ups by design do not require the wakeword, and, therefore, classifying them correctly is far more challenging (see Fig. 1).

Previous approaches to DDSD process isolated utterances (i.e., no previous context is considered) either directly from audio [2], text [6], or from intermediate Automatic Speech Recognition (ASR) lattice-based features [7]. A few recent approaches attempted classification of device-directed speech in the context of natural turn-taking [3; 4; 8] by exploring various acoustic and lexical features; however, these works do not account for joint modeling of the ASR uncertainty and multi-turn user queries. The approach proposed here is built upon the recent work in [1] that focuses on the DDSD

*Equal Contribution. Correspondence: *{orudovic,yi_su}@apple.com*

38th Conference on Neural Information Processing Systems (2024)-*Workshop on Adaptive Foundation Models*

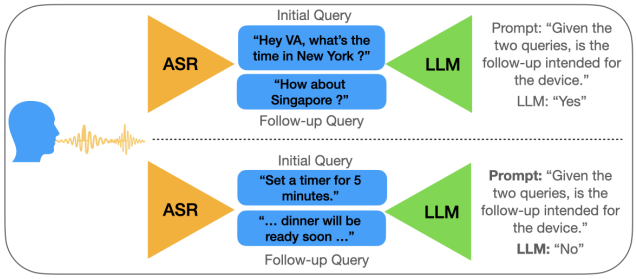

Figure 1: Follow-up conversations: the pair of user queries are first processed by an ASR system, which outputs the text transcriptions of the user's speech. The joint ASR transcription of the initial and follow-up queries are input to the LLM that detects if the latter is directed to the Virtual Assistant (VA).

Prompt: In this task, we provide a pair of queries made by human in the following format: 'Query 1: <text> | Query 2: <text>'. Query 1 is directed toward the voice assistant. Query 2 is the follow-up query made by human. {*For Query 2, we provided an n-best list of ASR hypotheses for the spoken utterance. Each of the hypothesis is separated by a newline character. The cost of each hypothesis is at the end in the format '[cost]' where a low cost indicates that we are more confident about that ASR hypothesis.*} Determine whether Query 2 is directed towards a voice assistant or a human being. Typical spoken utterances directed towards the voice assistant are commands to fulfill a task or queries to get some information. Answer only from the following categories ['1', '0'] where '1' indicates that the utterance is directed towards the voice assistant and '0' indicates that the utterance is directed towards a human being. In your answer the last line should contain nothing else but the number '0' or '1'.

Figure 2: Task-prompt used for Device-directed Speech Detection. Note that for the follow-up query, the text in *italics* is additionally added when including the n-best hypothesis by the ASR system.

task from a single query using a combination of off-the-shelf ASR model (n-best ASR hypotheses) and LLM. We focus on a more challenging task – DDSD for follow-ups, where, given the previous query of the user, the goal is to determine if the follow-up query is directed to the device. For this, we propose two LLM-based approaches: (i) prompting-based, where we experiment with direct *text-prompting* of a pretrained, and a finetuned LLM (Sec. 2.1). (ii) classification-based approach, where, instead of parsing the LLM decisions from output text, as in (i), we introduce a classification head on top of the LLM, so the approach can make probabilistic binary decisions for the DDSD (Sec. 2.2). Specifically, we simultaneously address two key limitations of the existing systems for DDSD: (i) the lack of context – where only the single query is modeled; instead, we account for both the initial and follow-up queries, thus, providing valuable information to the LLM when making decisions, and (ii) ASR uncertainty – while ASR models aim to transcribe speech to text accurately, speech recognition from the real-world speech is often inaccurate and the 1-best ASR hypothesis acts as an information bottleneck between the ASR and the LLM component. We expand this information bottleneck by exposing the LLM to an n-best list of ASR hypotheses, and we do so only for the follow-up utterance to prevent confusing the model, i.e., we focus on the uncertainty in the follow-up only. Apart from direct prompting, we also explore finetuning of Low-Rank Adaptation (LoRA) [9] adapters using training data of the n-best list prompts. We use an in-house general-purpose English ASR system and Vicuna [10], an instruction-tuned LLaMA LLM [11].

Our experiments show noticeable gains in terms of the DDSD accuracy over the simple approach where only the follow-up query is used as input to LLM. While this is intuitive, we empirically show on the challenging real-world dataset of follow-up conversations that jointly modeling the context (the previous and follow-up query), together with ASR uncertainty, helps to improve the LLM detection accuracy in the range of $\sim$20-40%, when the classifier is trained on top of the LLM. It also largely outperforms the traditional prompting-based approach without the LLM tuning on the target task.

2 Methods

The goal is to identify whether a spoken follow-up utterance is directed towards a device (e.g. smartphone) or a human (see Fig. 1). To prepare the input for the LLM, we take the ASR outputs

that come in the form of 1-best and n-best hypotheses, derived from a *lattice* graph comprised of the competing ASR hypotheses under a beam-search decoder [12]. The least cost path in the lattice is the 1-best hypothesis, while the n-best list comes from the full lattice (obtained by picking the n least cost paths in the lattice), which better captures the ASR uncertainty. We describe the follow-up query with either 1-best or n-best hypotheses whereas the initial query is only described using the 1-best hypothesis. We concatenate these ASR information from the initial and follow-up query to obtain the *"utterance-prompt"* (see Table 1). In the n-best lists of the follow-ups, the hypotheses are separated with newline characters and each hypothesis is appended with a *hypothesis-cost* at the end in the format *[cost]*. This cost is the sum of the acoustic- and language-model costs on the arcs along the lattice-path corresponding to the hypothesis [13], where a low cost indicates a high posterior probability assigned by the ASR [1].

utterance-prompt	Initial	Follow-up
1-best	*"Hey VA, play music"*	*"turn it up a bit"*
n-best	*"Hey VA, play music"*	*"turn it up a bit [-81.4]* *turn it up a bet [-78.1]* *term it up a pit [-75.9]*
Ground-truth	*"Hey VA, play music"*	*"turn it up a bit"*

Table 1: Examples of 1-best vs. n-best lists utterance-prompts, with the hypothesis cost shown in the brackets.

2.1 Prompting-based approach

In Fig. 2, we show an example of the fixed prompt, referred to as the *task-prompt*, that varies based on whether the 1- or n-best hypotheses are used in the follow-up. This task-prompt is further concatenated with the utterance-prompts containing the prompt with a follow-up. We consider two inference strategies: direct prompting of the pretrained LLM ("PromptOnly"), and prompting of a finetuned LLM ("FinetunePrompt") where the base LLM is finetuned using Parameter-Efficient FineTuning (PEFT) methods [14], specifically LoRA [9] - see Sec. 3.2.

2.2 Classification-based approach

Another way to take advantage of the pretrained LLM is to add a classifier head on top of it, and then to finetune the entire model (we refer to this approach as "ClassHead"). This approach is popularized by BERT [15] in prior literature. One of the drawbacks is that fine-tuning the entire LLM along with the classifier head not only requires a large amount of in-domain training data but also makes it difficult to deploy the model in production. To this end, we use the LoRA adapter finetuning again in the context of adding the LLM with a classifier head. This approach is referred to as "ClassLoRAHead". The classifier head in both "ClassHead" and "ClassLoRAHead" approaches is simply a linear layer that maps the last hidden layer output to a one-hot vector representing the device-directed speech label. In our task, the linear layer has 4096-D input, and 2-D output. The standard cross-entropy loss is used to train the LLM (or its LoRA adapters) and the classifier head.

3 Experiments

3.1 ASR System and Dataset

The ASR model that generates the 1- and n-best hypothesis for the input queries is based on the E2E-ASR architecture in [16], and it comprises of a Conformer [17] encoder with a CTC and attention-based decoder. The beam-search decodings from the attention-based decoder are rescored using an external Finite State Transducer-based language model (FST-LM). The ASR lattices obtained from the FST-LM decoding are used to generate the ASR hypotheses. Specifically, we use an internal audio dataset from demographically diverse English native speakers with consent, which is tailored to simulate a naturally occurring multi-turn conversations towards solving a task together (e.g, cross-word puzzles). The participants ask the VA questions, and use the answers to complete the task at hand, while the VA-equipped devices record audio from the participants. The dataset consists of $\sim$19k audio recordings from 1.3k participants. The conversations are clipped into roughly 245k segments, i.e., the pairs of utterances, where the first utterance is always device-directed, and the follow-up can be either directed or undirected. These data are human-annotated in terms of device-directness, and are split into training, validation, and test (70/10/20%) partitions, with no speaker overlap. The ratio of the device-directed and -undirected examples is $\sim$1:4, with average audio duration of 3.5$\pm$3.25 (mean$\pm$std) seconds.

Table 2: Accuracy of the prompting- and classification-based LLM approaches (in %). (*left*) The upper / lower rows show results for the follow-up query only (no context) / and pairs of queries (context), with the number of the ASR hypotheses (n) in the first column. (*right*) LLM models with a classification head. The paired t-test with 95% confidence intervals reveals statistically significant improvement (*) when comparing the last two rows (while both approaches bring statistically significant improvements when context is used, n=1-1 and 1-8).

n	PromptOnly		FinetunePrompt		ClassHead		ClassLoRAHead	
	FRR ↓	FAR ↓	FRR ↓	FAR ↓	FRR ↓	FAR ↓	FRR ↓	FAR ↓
1	**13.8**	**46.7**	12.6	16.4	20.6	2.0	19.5	2.4
8	37.2	56.7	18.6	2.3	17.7	1.6	17.5	2.2
1-1	20.1	74.7	18.6	2.3	16.3	2.0	17.6	2.2
1-8	65.2	33.2	**17.7**	**2.1**	**16.2**	**2.0**	**14.9**	**1.8**

n	ClassHead			ClassLoRAHead		
	EER %	FAR @5% FRR	FAR @10% FRR	EER %	FAR @5% FRR	FAR @10% FRR
1	10.5	29.9	11.4	10.6	30.1	12.7
8	9.5	25.6	8.3	9.6	30.5	8.7
1-1	8.5	18.3	6.1	8.9	19.9	7.3
1-8	**7.9**	**16.7**	**4.8***	**8.5**	**19.5**	**6.1***

3.2 LLM Finetuning

We use a pretrained instruction-tuned LLM, Vicuna-7B-v1.3 [10] in the experiments. The inference is done using 4 NVIDIA A100 GPUs. For finetuning the prompting-based approach, we train parameters of the LoRA adapters [9] for 3 epochs using 8 GPUs with a learning rate of 2e-5, with a warmup to over 3% of the learning steps. The LoRA adapters (rank=8) have 4.1M parameters that is only 0.06% of the 7B parameters of the used LLM. In the classification-based approach, we experiment with the full finetuning, and with LoRA adapters. We experimented with different number of training epochs, and we trained the former for 1 epoch, while for LoRA we use 3 epochs, as those were the best on the validation set (we found it to be less prone to overfitting). The classifier head adds only $\sim$8k additional parameters (the rest are the same as in the prompting-based approach). We use FastChat toolkit [18] for inference and finetuning with DeepSpeed GPU optimization [19], with 0 temperature during inference. Finetuning is done on *train/val* parts of the used dataset.

3.3 Evaluation Procedure

We evaluate the proposed approaches for the following setups. Firstly, we quantify the impact of the ASR uncertainty on classification of the follow-up query only (thus, no context), by representing it with 1- and n-best hypotheses. We set $n = 8$, found to work best on the validation data. Secondly, we quantify the impact of adding the context via the initial query, for which we only consider 1-best hypothesis (for the follow-up we compare both). We compare both the prompting- and classification-based approaches (see Table 2). Since the prompting-based approach produces only the textual binary outputs ("0" or "1" tokens), we compare in terms of False Accept Rate (FAR) and False Reject Rate (FRR) metrics, where the lower these metrics the higher the system accuracy. However, this approach does not enable to tune the DDSD system in practical applications as we cannot set target operating points (OP) (the binary labels do not allow to define the regime under a desired FRR). On the other hand, the classification-based approach is tunable as it provides probability scores. We report Equal Error Rate (EER) and FAR at hypothetical OPs 5% and 10% FRR. This allows us to establish the model accuracy across regimes relevant for user experience.

4 Results and Discussion

The prompting-based approach exploits the generative power of the LLM to *understand* the task based on the *prompt*, and outputs an answer that is then parsed to obtain the binary prediction for the speech being device-directed. We prompt the LLM system with the *task-prompt* and the *utterance prompt*, as described in Sec.2.1, without any finetuning ("PromptOnly"). We confirm empirically that the LLM can indeed perform the task out-of-the-box, but sometimes it outputs a descriptive answer in natural language, as opposed to the binary output (as desired for the task). To deal with this, when the LLM outputs a descriptive answer different from "0"/"1" tokens, we consider the follow-up utterance to be a device-directed speech (instead of removing the target utterance, which affects $\sim$6% of our data). The finetuned LLM ("FinetunePrompt") outputs only "0"/"1" values.

Table 2 (*left*) compares the prompting- vs. classification-based approaches. When prompting the LLM without finetuning ("PromptOnly"), the model performs best with the simple $n = 1$ hypothesis and without context. This is expected since the model is not finetuned on the target task, also suggesting that the LLM is not familiar with the concept of n-best lists from its original training, and, thus, it is unable to leverage the ASR uncertainty effectively (using the prompt design proposed here). The

model becomes confused by additional information (coming either from the ASR uncertainty in the follow-up prompt, and/or context from the initial query), and such a system is not useful in practice (note the high FAR of 46.7%). On the other hand, when we LoRA finetune the prompt-based model ("FinetunePrompt") to output binary targets, the accuracy improves considerably. Note a reduction in FAR: when using the 1-best hypothesis, FAR reduces 46.7% $\rightarrow$ 16.4%. Interestingly, by using the $n = 8$-best hypotheses (without context) brings the FAR to as low as 2.3%. While this comes at the expense of the increase in FRR (12.6% $\rightarrow$ 18.6%), it still renders a model that can be useful in practice. By looking at the role of the context, we note that adding the n-best hypothesis for the initial query helps when the n-best hypothesis is used for the follow-up (FAR 16.4% $\rightarrow$ 2.3%). This shows the importance of the context. However, the cumulative improvement in accuracy is reduced when both the context and ASR uncertainty are used jointly in this experiment. We also observed diminishing gains as we increase the size of the n-best list beyond $n = 8$, which can be due to the model overfitting and/or getting more confused by the uncertainty from too many ASR hypotheses. As observed in [1], LoRA finetuned prompting-based approach ("FinetunePrompt") does not require a task prompt since the model focuses on the target label, given the utterance prompt. On the other hand, the full classification finetuning was robust to whether the prompt was used or not in this task.

Next, we compare the classification-based approaches ("ClassHead" and "ClassLoRAHead"), where a linear classifier (head) is added on top of the base LLM (see Sec. 2.2). To compare with their prompting-based counterparts, for fairness, we binarize the classifier probability scores with 0.5 threshold. First, we see that "ClassHead" is able to leverage the ASR uncertainty (no context), with reduction in FAR (2.3% $\rightarrow$ 1.6%). By adding context, we note the reduction in FRR, however, it increases FAR (1.6% $\rightarrow$ 2.0%). Overall, these results suggest that the full finetuning of the classifier on top of the LLM does not benefit from additional context in this regime (with the used threshold). However, despite similar accuracy to that of "FinetunePrompt", the "ClassHead" approach is still a preferred candidate for the real-world usage as it can be tuned to achieve desired user experience (as it outputs probabilistic scores). Note that the LoRA adapters in the classification-based approach ("ClassLoRAHead") are able to efficiently use both, the ASR uncertainty and previous context, with the best achieved FRR of 14.9% and FAR of 1.8%. This approach is also easier and faster to train, as only a fraction of the model parameters need to be updated.

Table 2 (*right*) shows the results by the classification-based approach at different regimes using its probabilistic output. Specifically, we report for two exemplary OPs: FRR of 5% and 10%, and measure the EER/FAR that the system would produce under these false rejections, allowing us to directly compare the two classification models ("ClassHead" and "ClassLoRAHead"). Overall, "ClassHead" approach is more accurate, as it allows to tune all model parameters. The results support our hypothesis that modeling both, the ASR uncertainty and context of the previous query, *jointly* is critical for improving the model accuracy ($\sim$20-40% reduction in FAR at 10%FRR, when comparing the rows with n=8 and n=1-8). To better understand the models' behaviour, we listened to a number of utterance pairs that were falsely accepted at 10% FRR by models without context modeling (rows 2-3), but rejected when context is used (rows 4-5). Since there was no a continuation of the first request, the context (i.e. the initial request) helped to disambiguate the decision for the follow-ups. We observed a similar trend for the false rejects, where the context helped to accept the follow-ups as device-directed speech. To quantify the overall robustness, we report Detection Error Trade-off (DET) curve in Supp. Materials.

5 Conclusions

We explored various techniques for adapting an LLM model, including prompt tuning, and fine-tuning strategies. We showed that by integrating knowledge from a previous request of the user can largely improve the accuracy of device-directed speech detection in the follow-ups. These are often unconstrained in the content, and may not always be directed to the device, therefore, classifying them accurately is important for enabling naturalistic user experience and engagement. Specifically, we showed that the prompting-based approach leads to high accuracy when the LLM is fine-tuned on the DDSD task, however, it exhibits poor performance when *only* the engineered prompts are used (despite of adapting the language content in the prompts). On the other hand, adding a classifier on top of the pretrained (frozen) LLM allows the model to focus on the contextual clues from the first (initial) query, as well as to leverage effectively the ASR uncertainty (via the n-best hypotheses) – both encoded in the prompts, when making decisions about the follow-ups. We also showed that the classifier-adapted approaches can achieve high accuracy by tuning a fraction of the LLM parameters (via LoRA), saving the compute time. While in this work we focused on the pairs of the user queries as input to the LLM, other signals (e.g. VA's responses, acoustic features, and speaker information) can also be integrated to further improve the system accuracy.

References

[1] P. Dighe, Y. Su, D. Zheng, Y. Liu, V. Garg, X. Niu, and A. Tewfik, "Leveraging large language models for exploiting asr uncertainty," in *ICASSP*, 2024.

[2] 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.

[3] S. H. Mallidi, R. Maas, S. Matsoukas, and B. Hoffmeister, "Device-directed utterance detection," in *Interspeech 2018*, 2018.

[4] 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*, 2020.

[5] T. N. Sainath and C. Parada, "Convolutional Neural Networks for Small-Footprint Keyword Spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] D. Wagner, A. Churchill, S. Sigtia, P. Georgiou, M. Mirsamadi, A. Mishra, and E. Marchi, "Multimodal data and resource efficient device-directed speech detection with large foundation models," in *ICASSP, NeurIPS Workshop*, 2023.

[7] R. Agarwal, X. Niu, P. Dighe, S. Vishnubhotla, S. Badaskar, and D. Naik, "Complementary language model and parallel bi-lrnn for false trigger mitigation," in *Interspeech*, 2020.

[8] J. Wang, L. Chen, A. Khare, A. Raju, P. Dheram, D. He, M. Wu, A. Stolcke, and V. Ravichandran, "Turn-taking and backchannel prediction with acoustic and large language model fusion," in *ICASSP 2024*, 2024.

[9] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[10] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[11] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023.

[12] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[13] 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.

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

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805 [cs]*, 5 2019.

[16] D. Wu, B. Zhang, C. Yang, Z. Peng, W. Xia, X. Chen, and X. Lei, "U2++: Unified two-pass bidirectional end-to-end model for speech recognition," *arXiv preprint arXiv:2106.05642*, 2021.

[17] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[18] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[19] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters," in *ACM SIGKDD*, 2020.

A Supplemental material

To quantify the overall robustness of the models, we report Detection Error Trade-off (DET) curve in Fig. 3. We see a similar trend across most of the regimes, with both ASR uncertainty and context of the previous query in the classification-based approach performing the best, and also outperforming the prompting-based approach ("FinetunePrompt" with LoRA adapters, depicted in *green*). Finally, while the LoRA-based approach regresses compared to the full finetuning, with the largest regression occurring at 10% FRR, where FAR goes from 4.8% to 6.1%, both models still achieve relatively high accuracy. The LoRA-based approach has less tunable parameters and is much faster to train (on the used dataset, the reduce in the train time was ∼5-10 times). Therefore, a trade-off needs be made depending on the hardware and time resources.

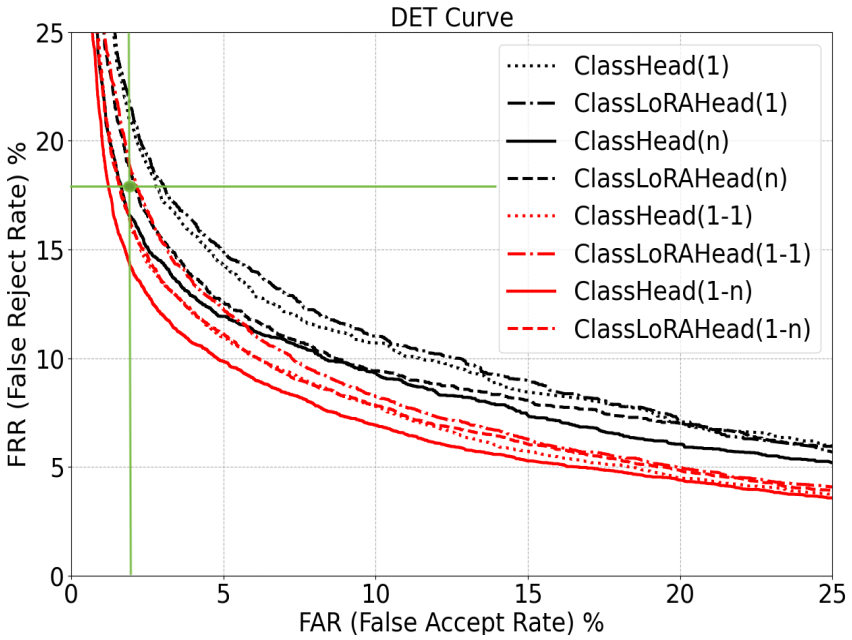

Figure 3: DETs showing the accuracy of the models across a range of regimes. Next to the model names in the brackets, we show the type of ASR hypothesis for each model reported (1- and $n = 8$,w found on the validation set to work best for the task). The lines in *black* and *red* represent the models without and with context of the previous query, respectively. The *green* dot shows the accuracy of the best prompting-based approach ("FinetunePrompt" with LoRA adapters).