INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A Study for Improving Device-Directed Speech Detection toward Frictionless Human-Machine Interaction

Che-Wei Huang, Roland Maas, Sri Harish Mallidi, Björn Hoffmeister

Amazon, USA

{cheweh,rmaas,mallidih,bjornh}@amazon

Abstract

In this paper, we extend our previous work on device-directed utterance detection, which aims to distinguish voice queries intended for a smart-home device from background speech. The task can be phrased as a binary utterance-level classification problem that we approach with a DNN-LSTM model using acoustic features and features from the automatic speech recognition (ASR) decoder as input. In this work, we study the performance of the model for different dialog types and for different categories of decoder features. To address different dialog types, we found that a model with a separate output branch for each dialog type outperforms a model with a shared output branch by a relative 12.5% of equal error rate (EER) reduction. We also found the average number of arcs in a confusion network to be one of the most informative ASR decoder features. In addition, we explore different frequencies of backward propagation for training the acoustic embedding for every k frames (k=1,3,5,7), and mean and attention pooling methods for generating an utterance representation. We found that attention pooling provides the most discriminative utterance representation and outperforms mean pooling by a relative 4.97% of EER reduction.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

Smart-home devices, such as the Amazon Echo and Google Home, are popular nowadays in many households all over the world. However, a smart-home device still faces challenges especially when the environment is surrounded by multiple background audio sources, for instance, a living room. In these scenarios, it can be crucial for the device to actively distinguish a voice query intended for the device (denoted device-directed) from background speech (denoted nondevice-directed).

In addition, with device-directed speech detection, the interaction between a device and a user would become more natural. Consider the following two dialogs:

Dialog 1

User: *Alexa, set a timer.*

Alexa: *For how long?*

User: *Five minutes.*

Dialog 2

User: *Alexa, what is the weather?*

Alexa: *Today in Seattle you can expect showers.*

User: *When will it rain today?*

Although these two cases are natural for humans, they pose different levels of challenges for the device. In the first case, the device would expect the dialog to continue on to the second-turn interaction, i.e. the user would respond with further information (*"Five minutes"*), since the device replies with a request. On the other hand, in the second case it is difficult for the device to know whether the user will continue the dialog because the device simply provides information in the first-turn interaction. A common solution to this scenario is to apply a wake-word for each interaction albeit it may add friction to the interaction. Therefore, as an alternative a device-directed speech classifier would help a human-machine dialog to continue with natural interactions. In this work, we denote the unexpected 2nd-turn utterance in Dialog 2 as a "2nd-turn" utterance for short, and all other utterances in Dialog 1 and 2 as "1st-turn" utterances. Please refer to section 3 for more details.

Our previous work [1] investigated the device-directed speech detection task and proposed a framework, similar to endpointing in [2], that integrates multiple information sources, including acoustics, speech decoding hypothesis and decoder features from an automatic speech recognition model, into one single device-directed model. For a given user query, we run ASR decoding to extract features, including one-best hypothesis, Viterbi costs, the average number of arcs in a confusion network and trellis entropy. The acoustics and decoding hypotheses are independently modeled by an LSTM to extract discriminative embedding vectors. A top-level model is trained to consume this pre-processed information, i.e. decoder features and embedding vectors, to decide whether an utterance is intended for the device or not.

In this paper, we present several improvements to our previously published model. First, our previous work pooled all these utterances together for modeling regardless of the information about the function of a dialog. We hypothesize that addressing each dialog type differently will provide further improvement to each dialog type and hence the combined utterances. A comparable example is multi-dialectical acoustic modeling. When facing multiple dialects in acoustic modeling, a model that disregards the variation in linguistic information may fail to generalize well [3, 4]. In addition, we also review other techniques for training acoustic embeddings. Previously, we have explored the direction of training with frame-level targets, where frame-level targets are derived from utterance-level targets. Inspired by a recent work [5], where acoustic embedding training is improved by incorporating global information within an utterance using frame-level attention, we aim to study other levels of granularity between frame-level and utterance-level utterance representations. Since a device-directed model takes decoder features as input, as the underlying speech recognition model changes, the corresponding decoder features change and hence the device-directed model. In this work, we also report an ablative study on the decoder features and the performance of the decoder features as ASR changes in order to gain more insight for future research.

Studies for device-directed speech detection used acoustic features and ASR decoder features [6, 7, 8, 9, 10]. Low level acoustic features such as energy, pitch, speaking rate and du-

http://dx.doi.org/10.21437/Interspeech.2019-2840

ration and their statistical functionals are computed for device-directed detection [7]. Other acoustic features such as multi-scale Gabor wavelets are studied in [8]. ASR decoder features, including confidence scores and N-grams are used in [7, 8]. Recently, a study [5] showed that attention mechanism can help to reduce EER based only on acoustic features.

This paper is organized as follow. First, we present an overview of the device-directed system in the next section, and layout the sections for the discussion of model components. In section 3, we propose our multi-dialog-type modeling to address different dialog types in a single model, followed by an ablative study for decoder features and their collective performance with respect to four ASR models. In section 5, we extend and explore topics to make acoustic embedding more discriminative including the type of acoustic features, and training and representation formulation techniques. The last section contains the conclusion.

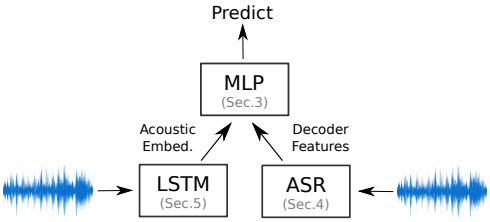


Figure 1: *An overview of the device-directed model*

2. System Overview

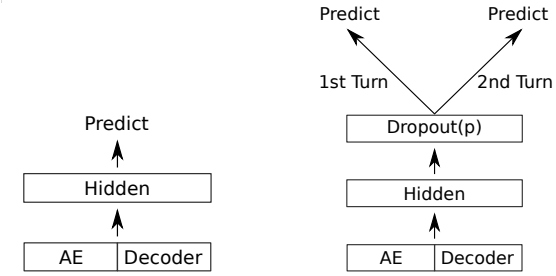
Fig.1 depicts an overview of the previously published device-directed system. A device-directed speech detection system in this work consists of decoder feature extraction from an ASR model, acoustic embedding extraction from an acoustic LSTM and a multi-layer perceptron (MLP) that takes these two sets of features as input, as indicated by the annotated arrows, to predict whether an utterance is device-directed. When a query is received, we feed it into the LSTM and cloud ASR to extract the acoustic embedding and decoder features. Once the feature extraction is done, the device-directed MLP is in charge to predict based on these features, whether this query is device-directed or not. The cloud ASR is a speech recognition system that is built with a hybrid HMM-LSTM acoustic model and a 4-gram language model.

In this work, we will study techniques to improve each of these three components in Fig.1 from section 3 to 5.

3. Dialog-Dependent Classification

A dialog type in this work refers to a concept to tag an utterance by its nature during a dialog between a smart-home device and a user. In this section, we focus on utterances in the unexpected 2nd-turn dialog type, e.g. *When will it rain today?* in Dialog 2, and other utterances in expected utterances dialog type. For convenience, we call them the 2nd-turn and the 1st-turn dialog types, respectively. A dialog-independent classification system for device-directed speech detection is designed to predict whether an utterance is device-directed regardless of their dialog types. Our previous work [1] has explored this direction and showed that it is simple and effective to capture device-directed information by pooling all training utterances together.

In this section, we aim to examine the validity of our hypothesis, that it is beneficial to address variation in paralinguistic information from one dialog type to another with a sophisticated architecture, and take it into device-directed modeling consideration. The setup for dialog-dependent device-directed model training is similar to what is specified in [1], where the input features include the acoustic embedding (AE) and a set of decoder features (Decoder), and we focus on the final fully-connected model for the experiments in this section.



(a) *Baseline: a shared output layer for both dialog types*

(b) *A separate output layer for each dialog type*

Figure 2: *Architectures for Dialog-Dependent Classification*

The baseline for our dialog-dependent device-directed model is a plain fully-connected model that is trained on the pooled training utterances as shown in Fig.(2a), where the input features are mapped to a hidden representation (Hidden) and then an output layer is in charge to predict if it is a device-directed speech based the hidden representation. To address each dialog-type separately, we propose an architecture to learn a separate output layer for each dialog type as shown in Fig.(2b). Similar to [4], this architecture learns a shared hidden representation by the front layers and models the dialog-type specific information via separate output layers. In this work, we simplify the modeling by limiting our discussion to 1st-turn and 2nd-turn utterances.

Except for the Dropout layer, the proposed architecture is actually a case of the weight- and bias-adaptation in [4] when the interpolation vector α is a two-dimensional one-hot vector for 1st-turn and 2nd-turn dialog types in the following formulation:

$$h_i = f\left(\left(W + \sum_{j=1}^{N} \alpha[j] \cdot U_j\right) \cdot h_{i-1} + B \cdot \alpha + b\right), \quad (1)$$

where the weight and bias for each output layer in Fig.(2b) are $W + \alpha[j] \cdot U_j$ and $B \cdot \alpha + b$ for $j = 1, 2$, respectively. The mathematical resemblance between Eq.(1) and separate output weights suggests the proposed architecture could be made more compact in the future work.

The dataset for training the multi-dialog-type model is summarized in Table 1, in which there are $324K$ 1st-turn utterances and $13K$ 2nd-turn utterances. We train both models on the pooled training utterances and perform model selection on the EER of pooled Dev utterances, where the equal error rate (EER) is defined as the average of false positive rate and false negative rate when these two rates are equal. Note that when training the proposed architecture, the dialog type information is used and only the corresponding path is forward and backward propagated.

In addition to the architecture in Fig.(2b), where the Dropout layer is shared by both output branches, we further lift

Table 1: *Dataset for Multi-Dialog-Type System*

	1st-turn	2nd-turn
Train	292K	11.6K
Dev	3K	122
Eval	29.4K	1.3K

Table 2: *Shared Dropout. Relative EER changes (%) with respect to the baseline for various dropout rates*

Dialog Type	Dropout(p)			
	0.00	0.25	0.50	0.75
Combined	-1.77	-3.54	-0.09	-2.59
1st-turn	-1.77	-2.68	-0.09	-3.45
2nd-turn	11.20	1.57	-5.88	3.31

* Model selection is performed based on the median Dev EER over five runs

Table 3: *Dropout Applied Only on 2nd-turn Output. Relative EER changes (%) with respect to the baseline for various dropout rates*

Dialog Type	Dropout(p)			
	0.00	0.25	0.50	0.75
Combined	-1.77	-3.54	-0.09	-2.59
1st-turn	-1.77	-2.68	-0.09	-3.45
2nd-turn	11.20	8.66	-12.50	12.40

* Model selection is performed based on the median Dev EER over five runs

the Dropout layer up and apply it only on the 2nd-turn output branch as a variant. The experimental results are presented in Table 2 and 3 and results are based on the median of the Dev performance over five runs. From Table 2, it shows that with an appropriate Dropout rate at $p = 0.50$, the proposed model can achieve relative 0.09% and 5.88% of EER reduction, respectively in 1st-turn and 2nd-turn over the baseline. From Table 3, if we lift the Dropout layer up to the 2nd-turn output branch, we could focus the regularization power on the lower-resourced branch, i.e. 2nd-turn output branch, and obtain a better improvement of relative 0.09% and 12.50%, respectively, for 1st-turn and 2nd-turn. From both of the proposed variants, the combined result also demonstrates a relative 0.09% of EER reduction.

4. Ablative Study for Decoder Features

Since the device-directed model takes decoder features as input, the performance of the device-directed speech detection inherently depends on the effectiveness of the decoder features. There are four categories of decoder features, including acoustic model cost (AM), language model cost (LM), the average number of arcs from each node in the confusion network (CN) and trellis entropy (TE). In this section, our first goal is to benchmark the combinations of these four categories of decoder features for device-directed modeling to identify the most informative decoder features.

From Table 4, it shows that the informativeness of each category can be ranked by CN $>$ TE $>$ AM $>$ LM for device-directed modeling, and collectively these four categories leads to a lowest EER compared to any subset of these categories.

The analysis above is carried out by fixing an ASR model. We can further compare the effectiveness of decoder features with respect to different ASR models. For this benchmark, we conduct device-directed modeling based on four ASR models, denoted by M1, M2, M3 and M4, in the increasing order of speech recognition performance.

Table 4: *Relative changes (%) with respective to decoder feature sets*

Feature Set	AUC	EER	ACC
AM	0.00	0.00	0.00
LM	-11.90	30.36	-10.39
CN	11.96	-32.57	13.06
TE	4.62	-12.34	5.00
AM+LM	4.95	-11.84	5.14
AM+CN	13.27	-34.00	13.69
AM+TE	9.21	-23.25	9.43
LM+CN	12.92	-34.54	14.00
LM+TE	8.00	-19.91	8.31
CN+TE	13.75	-35.74	14.42
AM+LM+CN	14.06	-36.99	14.97
AM+LM+TE	11.21	-28.48	12.03
AM+CN+TE	14.64	-38.21	15.49
LM+CN+TE	14.46	-38.15	15.44
AM+LM+CN+TE	15.35	-40.47	16.44

Table 5: *Relative changes (%) with respective to ASR models*

ASR Models	AUC	EER	ACC
M1	0.00	0.00	0.00
M2	0.22	-0.85	0.13
M3	0.33	-1.03	0.12
M4	0.55	-1.10	0.36

From Table 5, it shows that the device-directed modeling can benefit from an improved underlying ASR model.

5. Acoustic Embeddings

Our previous work showed that the acoustic embedding and decoder features provide complementary information for device-directed modeling, and joint modeling on both of them could outperform individual modeling by a large margin. In this section, we revisit the training of acoustic embedding and investigate ideas for further improvement, including types of acoustic feature, frequencies of backward propagation and methods of utterance representation generation.

5.1. LFBE and Log-STFT

Log-filterbank energy (LFBE) has been a popular acoustic feature type in speech related tasks for a long time including speech recognition, computational para-linguistics and speech event detection and so on so forth. Recently, learning with trainable convolutional or recurrent filters from raw acoustic input such as raw wave form or log energy of short-time Fourier transform (log-STFT) is gaining more and more momentum in machine learning community because deep neural networks have the outstanding capability to formulate task-specific representation given a sufficient amount of data.

For device-directed speech detection, we have shown that acoustic embeddings based on 64-dimensional LFBE (LFBE64) is able to perform well. Given promising results in speech recognition, we extend our acoustic embedding training to learn directly from log-STFT acoustic features.

256-dimensional log-STFT (log-STFT256) acoustic features are extracted by a window of 25 milliseconds with 10 milliseconds shift. An LSTM model is trained on log-STFT256

Table 6: *Relative changes (%) with respect to different input features*

Feature Set	AUC	EER	ACC
Decoder	0.00	0.00	0.00
Decoder+LFBE64	0.43	-29.54	6.31
Decoder+log-STFT256	0.48	-32.39	6.92

features with respect to frame-level device-directed targets, where the frame-level targets are obtained by repeating the utterance label. We use Adam optimizer with the default setting to minimize the cross-entropy loss. The pre-softmax output of the last frame of input utterance is taken as the acoustic embedding.

Table 6 summarizes the experimental results. The baseline model is trained only on the decoder features. With the addition of LFBE64 acoustic embedding, the relative EER reduction is close to 30% and is consistent with our previous result. The device-directed model that is based on log-STFT outperforms the LFBE64-based model by a relative 2.85% of EER reduction.

5.2. Backward propagation at every k frames

We have been using frame-level targets by repeating the utterance level target. Instead of training at each frame-level target, in this section we study the performance of training at a reduced frequency of backward propagation from every frame to every $k > 1$ frame. This approach essentially treats a segments of k frames as one unit and takes the last frame of the segment as the representation of the segment. Fig.3 shows the arrangement of targets in skip-frame training, where every k output from the LSTM is used to compute loss and the rest is simply ignored.

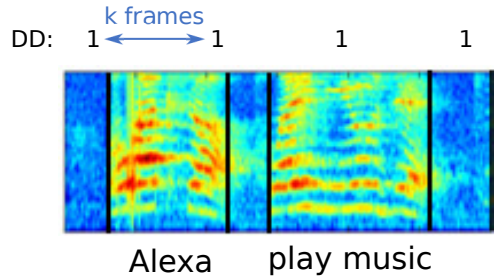


Figure 3: *Skip-k frames training*

5.3. Pooling Methods

In addition to training on frame-level or segment-level targets, since our true label is at the utterance-level, it makes perfect sense to generate an utterance-level representation and prediction, for the latter to be compared with the ground truth. For training with utterance-level target, we consider both mean and attention pooling for generating an utterance representation during training of acoustic embedding.

5.4. Experimental Setup and Results

So far, our acoustic embedding generation recipe is as follows: An LSTM is trained on every frame with frame-level targets derived from the utterance-level target. The acoustic embedding is extracted from the pre-softmax output of the last frame. We set this approach as the baseline and denote it as Train:Per-Frame, Eval:Last-Frame.

In the following, we conduct experiments to benchmark five other variants, including

a. Train:Skip 3 Frames, Eval:Last

b. Train:Skip 5 Frames, Eval:Last

c. Train:Skip 7 Frames, Eval:Last

d. Train:Mean-Pool, Eval:Mean-Pool

e. Train:Attention-Pool, Eval:Attention-Pool

Note we use log-STFT256 as the input feature in this section.

Table 7: *Relative EER changes (%) with respect to different training techniques*

Train, Eval	EER
Per-Frame, Last-Frame	0.00
Skip-3, Last-Frame	9.39
Skip-5, Last-Frame	4.97
Skip-7, Last-Frame	11.04
Mean-Pool, Mean-Pool	-30.39
Attention-Pool, Attention-Pool	-35.36

The experimental results are presented in Table 7. The attention-pooled model outperforms the mean-pooled model and the models trained with per-frame or skip-frame targets since an attention mechanism has access to and makes selective use of, as compared to mean pooling, the global information in generating a discriminative utterance representation. Training with skip-frame targets, however, does not improve from training with per-frame targets. Overall, an attention-pooled acoustic embedding outperforms an mean-pooled acoustic embedding by a relative 4.97% of EER reduction.

6. Conclusions

In this paper, we extended our previous work on device-directed modeling, and reviewed and explored several topics for improving the performance. We proposed to model multiple dialog types within a single model but address them differently with separate output layers, which is equivalent to the formulation for weight- and bias-adaptation training for multi-dialectical acoustic modeling. The experimental results show that using separate output layers is able to reduce EER compared to training with a shared output layer. In addition, the experimental results indicate that it preferable to apply regularization such as Dropout directly to where it is likely to overfit than to a shared representation. For example, instead of applying the Dropout layer to both dialog types, applying it only to the low resource dialog type further improves the performance. Through ablative study, we found that among four categories of decoder features, the average number of arcs in a confusion network is the most informative one, followed by trellis entropy, acoustic modeling cost and language modeling cost, and all of them collectively perform better than any subset of categories. We also showed that as the underlying ASR models improves, the device-directed modeling, which is based on the ASR models, can also benefit from the enhanced speech modeling power. We trained acoustic embedding based on log-STFT256 acoustic features and it outperforms the one based on LFBE64 acoustic features by a relative 2.85% EER reduction. We found that training for acoustic embedding with skipping frames does not help, compared to training with every frame, and training with an attention-pooled representation outperforms training with an mean-pooled representation.

7. References

[1] S. H. R. Mallidi, R. Maas, K. Goehner, A. Rastrow, S. Matsoukas, and B. Hoffmeister, "Device-directed utterance detection," in *Proceedings of Interspeech*, 2018.

[2] R. Maas, A. Rastrow, C. Ma, G. Lan, K. Goehner, G. Tiwari, S. Joseph, and B. Hoffmeister, "Combining acoustic embeddings and decoding features for end-of-utterance detection in real-time far-field speech recognition systems," in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing*, 2018.

[3] "Multi-dialectical languages effect on speech recognition: Too much choice can hurt," *Procedia Computer Science*, vol. 128, pp. 1 – 8, 2018, 1st International Conference on Natural Language and Speech Processing.

[4] M. Grace, M. Bastani, and E. Weinstein, "Occam's adaptation: A comparison of interpolation of bases adaptation methods for multi-dialect acoustic modeling with LSTMS," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, 2018, pp. 174–181.

[5] A. Norouzian, B. Mazoure, D. Connolly, and D. Willett, "Exploring attention mechanism for acoustic-based classification of speech utterances into system-directed and non-system-directed," in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing*, 2019.

[6] D. Reich, F. Putze, D. Heger, J. Ijsselmuiden, R. Stiefelhagen, and T. Schultz, "A real-time speech command detector for a smart control room," in *Proceedings of Interspeech*, 2011.

[7] E. Shriberg, A. Stolcke, D. Hakkani-Tr, and L. Heck, "Learning when to listen: Detecting system-addressed speech in human-human-computer dialog," in *Proceedings of Interspeech*, 2012.

[8] T. Yamagata, T. Takiguchi, and Y. Ariki, "System request detection in human conversation based on multiresolution gabor wavelet features," in *Proceedings of Interspeech*, 2009.

[9] H. Lee, A. Stolcke, and E. Shriberg, "Using out-ofdomain data for lexical addressee detection in humanhuman-computer dialog," in *Proceedings of HLT-NAACL*, 2013.

[10] D. Wang, D. Hakkani-Tur, and G. Tur, "Understanding computer-directed utterances in multi-user dialog systems," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.