Speech Processing for Digital Home Assistants

Reinhold Haeb-Umbach, Shinji Watanabe, Tomohiro Nakatani, Michiel Bacchiani, Björn Hoffmeister, Michael L. Seltzer, Heiga Zen, and Mehrez Souden

Abstract

Once a popular theme of futuristic science fiction or far-fetched technology forecasts, digital home assistants with a spoken language interface have become a ubiquitous commodity today. This success has been made possible by major advancements in signal processing and machine learning for so-called far-field speech recognition, where the commands are spoken at a distance from the sound capturing device. The challenges encountered are quite unique and different from many other use cases of automatic speech recognition. The purpose of this tutorial article is to describe, in a way amenable to the non-specialist, the key speech processing algorithms that enable reliable fully hands-free speech interaction with digital home assistants. These technologies include multi-channel acoustic echo cancellation, microphone array processing and dereverberation techniques for signal enhancement, reliable wake-up word and end-of-interaction detection, high-quality speech synthesis, as well as sophisticated statistical models for speech and language, learned from large amounts of heterogeneous training data. In all these fields, deep learning has occupied a critical role.

I. Introduction

In the last several years, the smart speaker has emerged as a rapidly growing new category of consumer electronic device. Smart speakers are internet-connected loudspeakers embodied with a digital assistant that can perform a variety of tasks through a hands-free spoken language interface. In many cases, these devices lack a screen and voice is the only input and output modality. These digital home assistants initially performed a small number of tasks, such as playing music, retrieving the time or weather, setting alarms, and basic home automation. Over time, the capabilities of these systems have grown dramatically, as developers have created third-party "skills" in much the same way that smart phones created an ecosystem of apps.

The success of smart speakers in the marketplace can be largely attributed to advances in all of the constituent technologies that comprise a digital assistant, including the digital signal processing involved in capturing the user's voice, the speech recognition that turns that audio into text, the natural language understanding that converts the text into a user's intent, the dialog system which decides how to respond, the natural language generation which puts the system's action into natural language, and finally the speech synthesis which speaks this response to the user.

In this article, we describe in detail the signal processing and speech technologies that are involved in capturing the user's voice and converting it to text in the context of digital assistants for smart speakers. We choose to focus in these aspects of the system since they are the ones most different from previous digital assistants that are resident on mobile phones. Unlike smartphones, smart speakers are located at a fixed location in a home environment, and thus need to be capable of performing accurate speech recognition from anywhere in the room. In these environments, the user may be several meters from the device and thus the captured speech signal can be significantly corrupted by ambient noise and reverberation. In addition, smart speakers are typically screenless devices, so they need to support completely hands-free interaction, including accurate voice activation to wake up the device.

We will give an account of the breakthroughs in the field of far-field automatic speech recognition (ASR), whereby reliable recognition is achieved despite significant signal degradations. We show how the deep learning (DL) paradigm has penetrated virtually all components of the system and has played a pivotal role in the success of digital home assistants.

It is worthwhile to note that several of the technological advancements that will be described have been inspired or accompanied by efforts in the academic community which provided researchers the opportunity to carry out comprehensive evaluations of technologies for far-field robust speech recognition using shared data sets and a common evaluation framework. Notably, the CHiME series of challenges [1,2], the REVERB challenge [3], and the ASpIRE challenge [4] were met with large resonance within the research community.

While these challenges led to significant improvements in the state of the art, they were focused primarily on speech recognition accuracy in far-field conditions as a criterion for success. Factors such as algorithmic latency or computational efficiency were not considered. However, the success of digital assistants in smart speakers can attributed to not just the system's accuracy but also

its ability to operate with low latency, creating a positive user experience by responding to the user's query with an answer shortly after the user stops speaking.

II. The Acoustic Environment in the Home

In a typical home environment, the distance between the user and the microphone on the smart loudspeaker is on the order of a few meters. There are multiple ways in which this distance negatively impacts the quality of the recorded signal, when compared to a voice signal captured on a mobile phone or headset.

First, signal attenuation occurs as the sound propagates from the source to the sensor. In free space, the power of the signal per unit surface decreases by the square of the distance. This means that if the distance between the speaker and microphone is increased from $2\,\mathrm{cm}$ to $1\,\mathrm{m}$ the signal will be attenuated by $34\,\mathrm{dB}$! In reality, the user's mouth is not an omnidirectional source and therefore, the attenuation won't be this severe. However, it still points to a significant loss of signal power.

Second, the distance between source and sensor in a contained space like living room or kitchen causes reverberation caused by multipath propagation. The wavefront of the speech signal repeatedly reflects off of the walls and objects in the room. Thus, the signal recorded at the microphone consists of multiple copies of the source signal, each with a different attenuation and time delay. This effect is described by the acoustic impulse response (AIR) or its equivalent representation in the frequency domain, the acoustic transfer function (ATF), and reverberant speech is often modeled as the original source signal filtered by the AIR.

An AIR can be broadly divided into the direct signal and early reflections (up to roughly the first $50\,\mathrm{ms}$), and the late reverberation, as shown in Fig. 1. While early reflections are actually known to improve the perceptual quality by increasing the signal level compared to the "dry" direct path signal, the late reverberation causes difficulty in perception both for humans and for machines because it smears the signal over time [5].

The degree of reverberation is often measured by the time it takes for the signal power to decrease to $-60\,\mathrm{dB}$ below its original value. This is referredto as the reverberation time and is denoted by T_{60}. Its value depends on the size of the room, the materials of walls, floor and ceiling, as well as the furniture. A typical value for a living room is between 300 and $700\,\mathrm{ms}$. Because the reverberation time is usually much longer than the typical short-time signal analysis window of 20 to $64\,\mathrm{ms}$, its effect cannot be adequately described by considering a single speech frame in isolation. Thus, the convolution of the source signal with the AIR cannot be represented

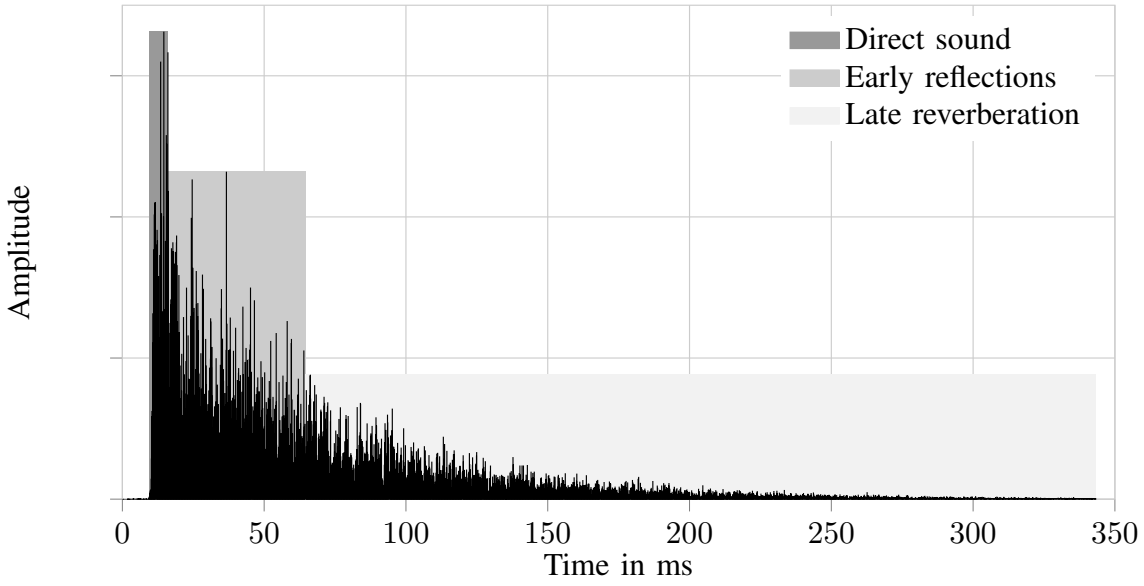

Fig. 1. An acoustic impulse response consists of the direct sound, early reflections and late reverberation

by a multiplication of their corresponding transforms in the short time Fourier transform (STFT) domain. It is instead approximated by a convolution over frames:

$$x_{t,f} = \sum_{m=0}^{M-1} a_{m,f} s_{t-m,f}. \tag{1}$$

Here, $x_{t,f}$, $s_{t,f}$, and $a_{t,f}$ are the STFTs coefficients of the reverberated signal, the source signal, and the AIR, respectively, at (discrete) time frame t and frequency bin index f. The length M of the STFT of the AIR is approximately given by T_{60}/B, where B is the frame advance (e.g., $10\,\mathrm{ms}$). Clearly, the effect of reverberation spans multiple consecutive time frames leading to a temporal dispersion of a speech event over adjacent speech feature vectors.

Third, in a distant-talking speech recognition scenario, it is likely that the microphone will capture other interfering sounds, in addition to the desired speech signal. These sources of acoustic interference can be diverse, hard to predict, and often nonstationary in nature and thus difficult to compensate. In a home environment, common sources of interference include television or radio, home appliances and other people in the room.

These signal degradations can be observed in Fig. 2. It shows signals of the speech utterance "Alexa stop" in a close-talk recording, a far-field recording, and a far-field recording with additional background speech. Clearly, keyword detection and speech recognition are much more challenging in the latter case.

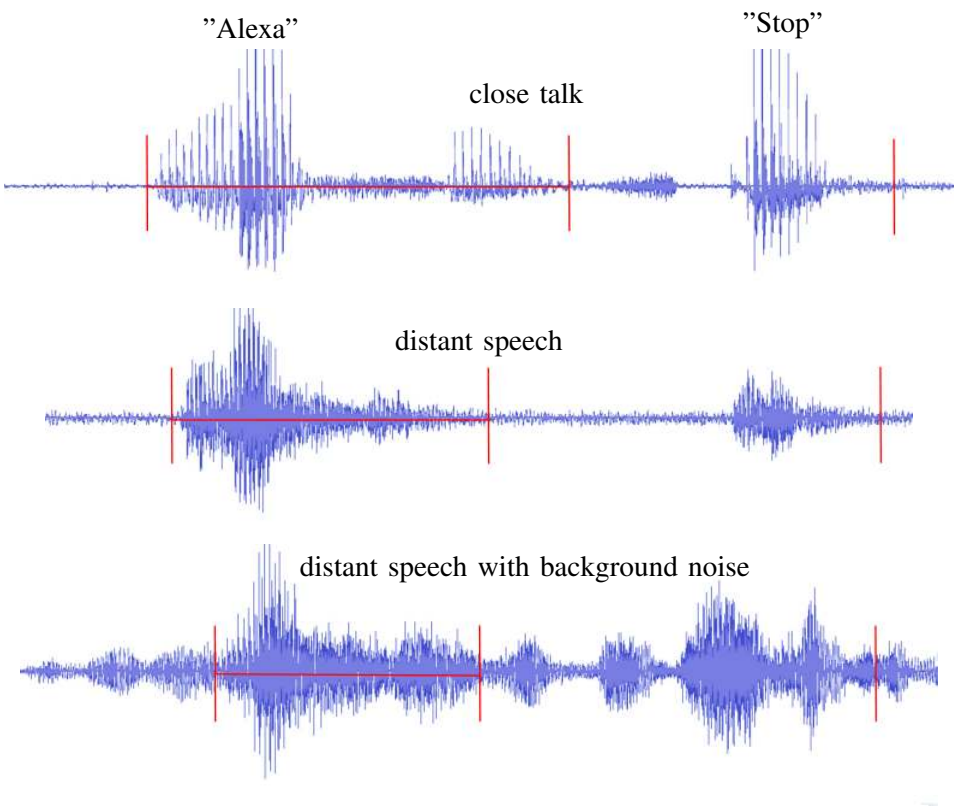

Fig. 2. Speech utterance starting with the wake word “Alexa” followed by “Stop” in close-talk, reverberated, and noisy reverberated conditions. The red bars indicate the detected start and end times of the keyword “Alexa” and the end of the utterance.

The final major source of signal degradation is the capture of the signals that originate from loudspeaker itself during playback. Because the loudspeaker and the microphone are co-located on the device, the playback signal can be as much as 30 to 40 dB louder than the user’s voice, rendering the user’s command inaudible, if no countermeasures are taken.

III. System Overview

Figure 3 gives a high-level overview of the speech processing components of a digital home assistant. For sound rendering, the loudspeaker system plays music or system responses. For sound capture, digital home assistants typically employ an array of microphones, typically 2 – 8. Due to the form factor of the device, the array is compact with distances between the microphones on the order of a few centimeters. In the following section, techniques from multi-channel signal processing will be described that can compensate for many of the sources of signal degradation described previously.

The signal processing front-end performs acoustic echo cancellation, dereverberation, noise reduction, and source separation, all of which aim at cleaning up the captured signal for input to the downstream speech recognizer. For a true hands-free interface, the system must detect whether speech has been directed to teh device. This can be done using

- a wake word (also called hotword, keyword, or voice trigger) detector, which decides if a user has uttered the keyword (e.g., "Ok Google") in order to address the device,
- an end-of-query detector, which is equally important to signal that the user's input is complete,
- a second-turn device-directed speech classifier, which frees the user from the need to again start with the wake word in an ongoing dialog, and
- a speaker identification module which makes the system capable of interpreting a query in a user-dependent way.

Once device-directed speech is detected, it is forwarded to the ASR component.

The recognized word sequence is then forwarded to the natural language processing (NLP) and dialog management subsystem, which analyzes the user input, and decides on a response. The natural language generation (NLG) component prepares the desired system response, which is spoken out on the device through the text-to-speech (TTS) component. Note, that NLP is beyond the scope of this article. The remainder of this paper is focused on the various speech processing tasks.

Some of the above processing tasks are carried out on the device, typically those close to the I/O, while others are done on the server. While the division between client and server may vary, it is common practice to run signal enhancement and wake word detection on the device, while the primary ASR and NLP processing are done on the server.

IV. Multi-Channel Speech Enhancement

The vector of the D microphone signals $\mathbf{y} = (y_1, \ldots, y_D)^{\mathrm{T}}$ at time frequency (tf) bin (t, f) can be written in the STFT domain as follows [6]:

$$\mathbf{y}_{t,f} = \underbrace{\sum_{i=1}^{N_\mathrm{s}} \sum_{m=0}^{M-1} \mathbf{a}_{m,f}^{(i)} s_{t-m,f}^{(i)}}_{\text{speech}} + \underbrace{\sum_{j=1}^{N_\mathrm{o}} \sum_{m=0}^{M-1} \mathbf{w}_{m,f}^{(j)} o_{t-m,f}^{(j)}}_{\text{playback}} + \underbrace{\mathbf{n}_{t,f}}_{\text{noise}}. \tag{2}$$

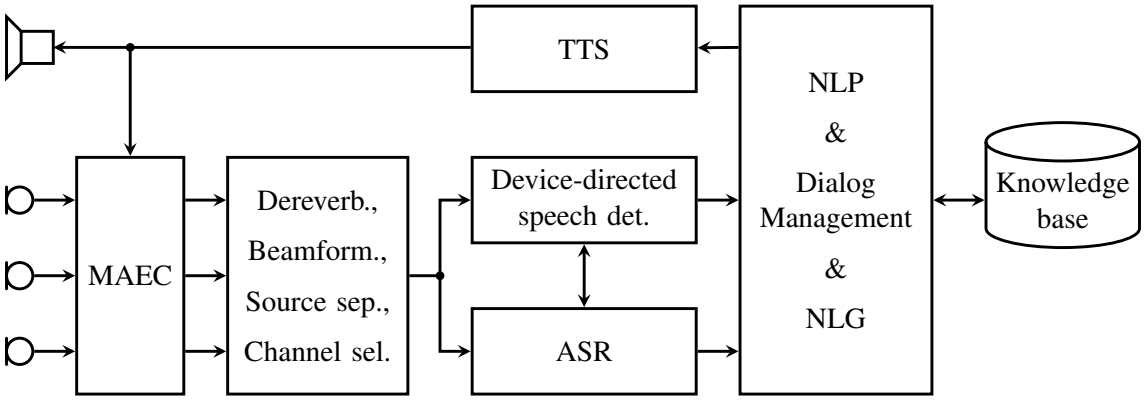

Fig. 3. Overview of example architecture of signal processing tasks in smart loudspeaker.

The first sum is over the N_s speech sources $s_{t,f}^{(i)}$, $i = 1, \ldots, N_\mathrm{s}$, , where $\mathbf{a}_{t,f}^{(i)}$ is the vector of ATFs from the i-th source to the microphones. The second sum describes the playback of the N_o loudspeaker signals $o_{t,f}^{(j)}$, $j = 1, \ldots, N_\mathrm{o}$, which are inadvertently captured by the microphones via the ATF vector $\mathbf{w}_{t,f}^{(j)}$ at frequency bin f. Further, $\mathbf{n}_{t,f}$ denotes additive noise. Here, we assume for simplicity that the transfer functions are time-invariant and of same length.

It is only one of the many signals, which contains the user's command, while all other components of the received signal are distortions. In the following we describe how to extract this desired signal.

A. Multi-Channel Acoustic Echo Cancellation

Multi-channel acoustic echo cancellation (MAEC) is a signal processing approach that is designed to prevent signals generated by a device's loudspeaker from being captured by the device's own microphones and confusing the system. MAEC is a well established technology, which relies on the use of adaptive filters [7]. Those filters estimate the acoustic paths between loudspeakers and microphones to identify the part of the microphone signal that is caused by the system output and then subtract it from the captured microphone signal.

Linear adaptive filters can suppress the echos by typically 10 to $20\,\mathrm{dB}$, but they cannot remove them completely. One reason is the presence of nonlinear components in the echo signal, which are caused by louspeaker nonlinearities and mechanical vibrations. Another reason is that the filter lengths must not be chosen to enable fast adaptation to changing echo paths. These lengths are usually shorter than the true loudspeaker-to-microphone impulse responses. Further, there is a well-known ambiguity issue with system identification in MAEC [7].

Therefore, it is common practice in acoustic echo cancellation to employ a residual echo suppressor following echo cancellation. In a modern digital home assistant its filter coefficients are determined with the help of a neural network [6]. The deep neural network (DNN) is trained to estimate, for each tf bin, a speech presence probability. Details of this procedure are described in the box entitled "Unsupervised and supervised speech presence probability (SPP) estimation" on page 12 for details. From this SPP a mask can be computed which separates desired speech dominated tf bins from those dominated by residual echoes, and from this information, the coefficients of a multi-channel filter for residual echo suppression are computed.

With MAEC in place it is possible that the device can listen to a command, while the loudspeaker is in use, e.g., for playing music. The user can barge in and still be understood, an important feature for user convenience. Once the wake-up keyword has been detected, the loudspeaker signal and the MAEC, are ducked or switched off, while the speech recognizer is activated.

B. Dereverberation

We now turn our attention to the first sum in Eq. (2). Assuming for simplicity that a single speech source is present, this term simplifies to Eq. (1).

As mentioned earlier, it is the late reverberation that is harmful to speech recognition performance. Decomposing the reverberated signal into the direct sound and early reflections $\mathbf{x}_{tf}^{(\text{early})}$ and the late reverberation $\mathbf{x}_{tf}^{(\text{late})}$ according to

$$\mathbf{x}_{tf} = \mathbf{x}_{tf}^{(\text{early})} + \mathbf{x}_{tf}^{(\text{late})}, \tag{3}$$

it is the late reverberation that a dereverberation algorithm aims to remove while preserving the direct signal and the early reflections.

There is a wealth of literature on signal dereverberation [5]. Approaches can be broadly categorized into linear filtering and magnitude or power spectrum estimation techniques. For ASR tasks, the linear filtering approach is recommended, because it does not introduce nonlinear distortions to the signal which can be detrimental to speech recognition performance.

Using the signal model of Eq. (1), where the AIR is a finite impulse response, a Kalman filter can be derived as the statistically optimum linear estimator under a Gaussian source assumption. Because the AIR is unknown and even time-varying, the Kalman filter is embedded in an Expectation Maximization (EM) framework, where Kalman filtering and signal parameter estimation alternate [8].

If the reverberated signal is modeled as an autoregressive stochastic process instead, linear prediction based dereverberation filters can be derived. A particularly effective method which has found widespread use in far-field speech recognition is the Weighted Prediction Error (WPE) approach [9]. WPE can be formulated as a Multiple Input Multiple Output (MIMO) filter, allowing further multi-channel processing, such as beamforming, to follow it [10,11]. The underlying idea of WPE is to estimate the late reverberation $\mathbf{x}_{tf}^{(\text{late})}$ and subtract it from the observation to obtain a maximum likelihood (ML) estimate of the early arriving speech:

$$\hat{\mathbf{x}}_{tf}^{(\text{early})} = \mathbf{x}_{tf} - \mathbf{G}_{tf}\tilde{\mathbf{x}}_{t-\Delta,f}. \tag{4}$$

Here, $\mathbf{G}_{tf}$ is a matrix containing the linear prediction coefficients for the different channels and $\tilde{\mathbf{x}}_{t-\Delta,f}$ are stacked representations of the observations: $\tilde{\mathbf{x}}_{t-\Delta,f} = (\mathbf{x}_{t-\Delta,f}^T, \ldots, \mathbf{x}_{t-\Delta-L+1,f}^T)^T$, where L is the length of the derverberation filter. It is important to note that $\hat{\mathbf{x}}_{tf}^{(\text{early})}$ at time frame t is estimated from observations at least Δ frames in the past. This ensures that the dereverberation filter does not destroy the inherent temporal correlation of a speech signal, which is not caused by the reverberation. The filter coefficient matrix cannot be estimated in closed form. The reason is that the driving process of the autoregressive model, $\mathbf{x}_{tf}^{(\text{early})}$, has an unknown and time-varying variance λ_{tf}. However, an iterative procedure can be derived which alternates between estimating the variance λ_{tf} and the matrix of filter coefficients $\mathbf{G}_{tf}$ on signal segments.

Because WPE is an iterative algorithm, it is not suitable for use in a digital home assistant, where low latency is important. However, the estimation of the filter coefficients can be cast as a Recursive Least Squares problem [12]. Furthermore, using the average over a window of observed speech power spectra as an estimate of the signal variance λ_{tf}, a very efficient low latency version of the algorithm to be used [13].

Many authors reported that WPE leads to word error rate (WER) reductions of a subsequent speech recognizer [13,14]. How much of a WER reduction is achieved by dereverberation depends on many factors such as degree of reverberation, signal-to-noise ratio, difficulty of the ASR task and robustness of the models in the ASR decoder, etc. In [13] relative WER improvements of 5% to 10% were reported on simulated digital home assistant data with a pair of microphones and a strong back-end ASR engine.

C. Multi-channel Noise Reduction and Beamforming

Multi-channel noise reduction aims at removing additive distortions, denoted by $\mathbf{n}_{t,f}$ in Eq. (2). If the acoustic impulse response from the desired source to the sensors is known, a spatial filter,

i.e., a beamformer, can be designed that emphasizes the source signal over signals with different transfer characteristics. In its simplest form this filter compensates for the different propagation delays that the signals at the individual sensors of the microphone array exhibit and that are caused by their slightly different distances to the source.

For the noisy and reverberant home environment this approach is, however, too simplistic. The microphone signals differ not only in their relative delay, the whole reflection pattern, they are exposed to, is different. Assuming again a single speech source and good echo suppression and dereverberation, Eq. (2) reduces to:

$$\mathbf{y}_{tf} = \mathbf{x}_{tf} + \mathbf{n}_{tf} \approx \mathbf{a}_f s_{tf} + \mathbf{n}_{tf}, \tag{5}$$

where $\mathbf{a}_f$ is the vector form of the AIRs to multiple microphones, and where we assume it to be time-invariant under the condition that the source and microphone positions do not change during a speech segment (e.g., an utterance)[1]. Note, that unlike Eqs. (1) and (2), the Multiplicative Transfer Function Approximation (MTFA) is used here, which is justified by the preceding dereverberation component. Any signal component deviating from this assumption can be viewed to be captured by the noise term $\mathbf{n}_{t,f}$. Similarly, residual echoes can be viewed to contribute to $\mathbf{n}_{t,f}$, resulting in a spatial filter for denoising, dereverberation and residual echo suppression.

Looking at Eq. (5) it is obvious that s_{tf} and $\mathbf{a}_f$ can only be identified up to a (complex-valued) scalar, because $s_{tf} \cdot \mathbf{a}_f = (s_{tf} \cdot C) \cdot (\mathbf{a}_f / C)$. To fix this ambiguity, a scale factor is chosen, such that for a given reference channel, say channel 1, the value of the transfer function is one. This yields the so-called relative transfer function (RTF) vector $\tilde{\mathbf{a}}_f = \mathbf{a}_f / a_{1f}$.

Spatial filtering for signal enhancement is a classic and well-studied topic, for which statistically optimal solutions are known. However, those text book solutions usually assume that the RTF $\tilde{\mathbf{a}}_f$, or its equivalent in anechoic environments, the vector of time differences of arrival (TDOAs), are known, which is an unrealistic assumption.

The key to spatial filtering is, again, SPP estimation, see the box on page 12 on this topic. The SPP tells us which tf bins are dominated by the desired speech signal and which by noise. Given this information, spatial covariance matrices for speech and noise can be estimated, from which in turn the beamformer coefficients are computed. An alternative is to use the SPP to derive a

[1]When the source (speaker) and/or microphone (sensor) are moving during a speech segment, we may need to track such position changes.

time-frequency mask, which multiplies tf bins dominated by noise with zero, thus leading to an effective mask-based noise reduction.

Figure 4 shows the effectiveness of beamforming for an example utterance. The figure displays the spectrogram, i.e., the time-frequency representation of a clean speech signal in subfigure (a), followed in (b) by the same utterance after convolution with an AIR, and in subfigure (c) after addition of noise. Subfigure (d) displays the output of the beamformer, which effectively removed noise and reverberation.

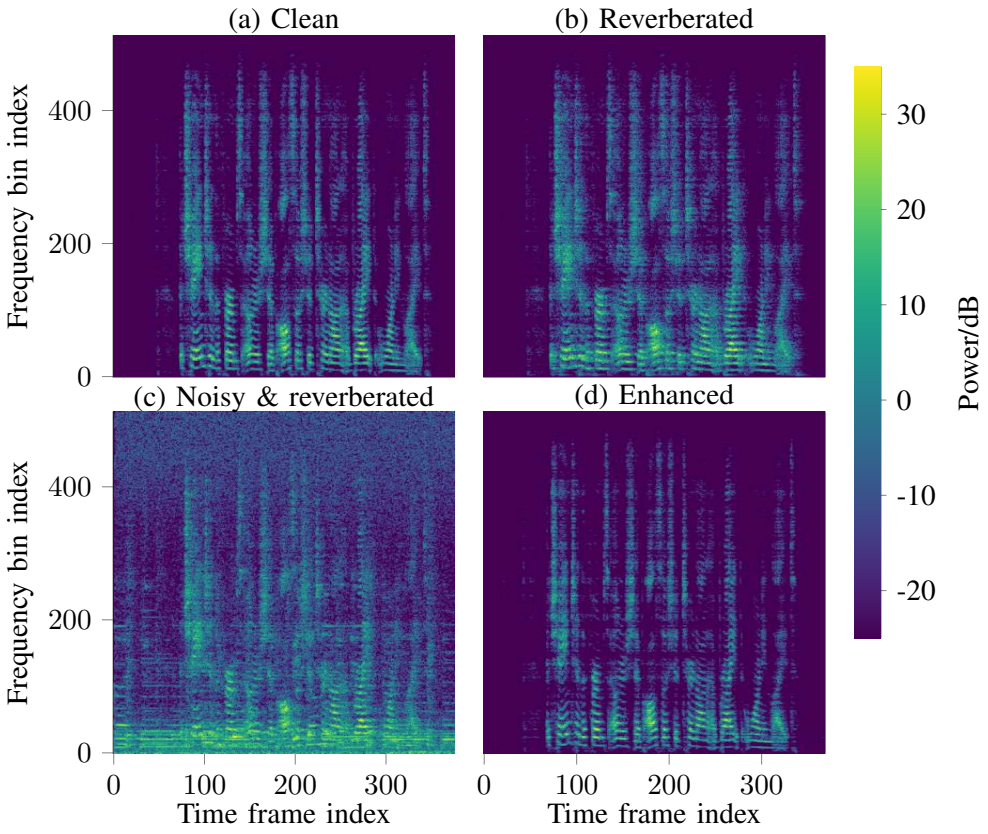

Fig. 4. Spectrogram of a (a) clean, (b) reverberated, (c) noisy and reverberated and (d) enhanced speech signal. Enhancement has been achieved with a beamformer which was trained to treat both noise and late reverberation as distortion.

The usefulness of acoustic beamforming for speech recognition is well documented. On the *Computational Hearing in Multi-source Environments (CHiME)* 3 and 4 challenge data, acoustic beamforming almost cut the word error rate in half. On typical digital home assistant data WER reductions on the order of 10% to 30% relative were reported [6,15,16].

Unsupervised and supervised speech presence probability (SPP) estimation

In the unsupervised learning approach, a spatial mixture model is used to describe the statistics of $\mathbf{y}_{tf}$ or a quantity derived from it:

$$p(\mathbf{y}_{tf}) = \sum_{k=0}^{1} \pi_k p(\mathbf{y}_{tf}|\boldsymbol{\theta}_k) \tag{6}$$

where we assumed a single speech source and where π_k is the a-priori probability that an observation belongs to mixture component k and $p(\mathbf{y}_{tf}|\boldsymbol{\theta}_k)$ is an appropriate component distribution with parameters $\boldsymbol{\theta}_k$ [17]–[19]. This model rests upon the well-known sparsity of speech in the STFT domain [20]

$$\mathbf{y}_{tf} = \begin{cases} \mathbf{a}_f s_{tf} + \mathbf{n}_{tf} & z_{tf} = 1 \\ \mathbf{n}_{tf} & z_{tf} = 0 \end{cases}, \tag{7}$$

where z_{tf} is the hidden class affiliation variable, which indicates speech presence. The model parameters are estimated via the EM, which delivers the SPP $\gamma_{tf} = \Pr(z_{tf} = 1|\mathbf{y}_{tf})$ in the E-step [21].

The supervised learning approach to SPP estimation employs a Neural Network (NN). Given a set of features extracted from the microphone signals at its input and the true class affiliations z_{tf} at the output, the network is trained to output the SPP γ_{tf} [22,23]. Since all STFT bins $f = 0, \ldots, F-1$ are used as input, the network is able to exploit inter-frequency dependencies, while the mixture model based SPP estimation operates on each frequency independently. If additionally cross-channel features, such as inter-channel phase differences, are used as input, spatial information can also be exploited for SPP estimation.

In a batch implementation, given the SPP the spatial covariance matrices of speech-plus-noise and noise are estimated by

$$\boldsymbol{\Sigma}_f^{(\mathbf{y})} = \sum_t \gamma_{tf} \mathbf{y}_{tf} \mathbf{y}_{tf}^H \Big/ \sum_t \gamma_{tf}; \quad \boldsymbol{\Sigma}_f^{(\mathbf{n})} = \sum_t (1-\gamma_{tf}) \mathbf{y}_{tf} \mathbf{y}_{tf}^H \Big/ \sum_t (1-\gamma_{tf}). \tag{8}$$

From these covariance matrices the beamformer coefficients of most common beamformers can be readily computed [21]. By an appropriate definition of the noise mask, this concept can also be extended to noisy and reverberant speech, leading to a significant dereverberation effect of the beamformer [24], see the example in Fig. 4.

For use in a smart loudspeaker low latency is important, which impacts both the design of the EM (or statistical methods in general) and the NN based approaches, see, e.g., [6,15,25] for a discussion.

D. Source Separation and Stream Selection

Now we assume that in addition to the desired speech source there are other competing talkers, resulting in a total of N_s speech signals, see Eq. (2).

Blind source separation (BSS) is a technique that can separate multiple audio sources into individual audio streams in an unsupervised fashion. Traditionally, researchers tackle speech source separation using either unsupervised methods, like independent component analysis and clustering [26], or deep learning [27,28]. In the particular case of clustering, BSS using spatial mixture models is a powerful tool to decompose the microphone array signal into the individual talkers' signals [17]–[19]. The parameters and variables of those mixture models are learnt via the EM algorithm as explained in the box. The only difference is that the mixture model now has as many components as there are concurrent speakers. During the EM, for each speaker a source activity probability (SAP), which is the equivalent to the SPP in the multi-speaker case, is estimated.

Extraction of the individual source signals can be achieved by using the estimated SAP to derive, for each speaker, a mask, by which all tf bins not dominated by this speaker are zeroed out, so-called mask-based competing speaker suppression and noise reduction. An alternative is to use the SAP to compute beamformers, one for each of the speakers, similar to what is explained in the box.

Once the sources are separated, it remains to be decided, which of the streams contains the user's command for the digital home assistant. In [6] it is proposed to base this decision on the detection of the wake-up keyword (e.g., "Hey Siri"): If the wake word detector indicates presence of the keyword, all streams, i.e., the output streams of source separation and the output of the acoustic beamformer, are scored for the presence of the keyword, and the stream with the highest score is considered to contain the user's command.

V. Automatic Speech Recognition

The key knowledge sources for ASR are the acoustic model (AM), the pronunciation model and the language model (LM). The language model assigns probabilities to hypothesized strings. The pronunciation model maps strings to subword units, where the phoneme is a common choice. Probabilities of the acoustic realization of the subword units (generally using some context) are expressed by the AM.

The AM of a speech recognition system is realized by a deep neural network. Such models estimate the posterior probabilities of subword units in context given the input signal. State-

of-the-art network architectures borrow architectural concepts from image recognition networks, e.g., the ResNet [29]. They will also include sequential modeling through the use of recurrent structures. Many sites use long short-term memory (LSTM) network layers or time delay neural network (TDNN) structures to incorporate that temporal component into the model.

The AM is trained from examples and generally requires large corpora to allow robust parameter estimation of these models (on the order of thousands of hours). It is essential that these corpora reflect the type of utterances that the device will recognize. For very novel applications, as was the case with the early deployment of digital home assistants, example data was not available, and collecting such large amounts of training data before product launch was considered uneconomic. Complicating matters even more, the expected variability is very large for speech coming into such a device. Therefore, the bootstrapping problem of a model for a digital home assistant is considerably more complex than for other new application domains. Approaches to this bootstrap problem are discussed below.

A. Bootstrapping the acoustic model

Most sites that developed the early digital assistants have large data sets of in-domain, close-talking material available. To make use of that data but render it suitable for the digital home assistant application, simulation techniques are employed. Using the well-known image method [30], sufficiently realistic AIRs can be generated for given room and microphone parameters. Alternatively, measured AIRs can be used, such as the collection in [31]. It is of course much easier to simulate thousands of AIRs representing large varieties of room and microphone array configurations in this way than to measure them. The non-reverberant close-talk recordings are then convolved with these AIRs to generate reverberant speech. It should be mentioned, though, that this simulates a static scenario. In reality, an AIR is time-varying: even the smallest movements of the speaker or changes in the environment will lead to a different reverberation pattern. Nevertheless, experience tells that systems trained with artificially generated reverberant speech perform robustly on real reverberant data.

Additional data augmentation techniques are used [32] in the development of an AM for a new application, which perturb existing recordings along different perceptually relevant parameters, such as speaking rate, vocal tract length, pitch, signal-to-noise ratio, noise types, etc.

B. Integrating enhancement and acoustic modeling

Although experimentation showed that the simulation of reverberant distant-talking speech from close-talking copora was effective in mitigating a lot of the problems posed in this setting, there is a large body of work on using enhancement from multi-channel processing to (further) mitigate the problems that arise in distant-talking speech recognition, as discussed earlier. However, independent optimization of the enhancement component and acoustic modeling component might not lead to performance improvements per se since a mismatch in the training objectives can adversely affect the overall system performance. It appears to be advantageous to optimize the AM and enhancement component jointly with a criterion close the the ASR objective. This bypasses the signal-related objective functions, like maximizing the output SNR, used in classic beamforming to ensure the enhancement result is to the benefit of the ASR that consumes its output. This direction has been first advocated by [33] in Gaussian mixture based acoustic modeling. More recently, it has been proposed to perform multi-channel enhancement jointly with acoustic modeling in a DNN framework [34,35]. To leverage the differences in the fine time structure of the signals at the different microphones, it is necessary that the raw time domain signal or its equivalent complex-valued STFT representation is input to the network. This is different from standard acoustic modeling, where the time-domain signal is first compressed to feature vector representations, such as logarithmic mel spectra or cepstra, which no longer carry subtle time information. A close look at the filter coefficients learnt in the initial layers of the network showed that indeed beamformer-like spatial filters could be identified and that frequency resolutions resembling the mel filterbank were found [34].

An alternative to this single large enhancement and acoustic modeling network is to keep enhancement and AM separate and still optimize both jointly towards an ASR-related criterion. It has been shown in [36] how the neural network for SPP estimation, see the box on page 12, can be trained from the objective function of the acoustic model by back-propagating the gradient through the AM DNN and the beamforming operation all the way to the DNN for SPP estimation. Clearly, this can be viewed as one large DNN with fixed, non-trainable signal processing layers in-between.

A direct comparison of the fully integrated approach with the separate, however jointly trained, speech enhancement and acoustic modeling stages on a common ASR task is not known. Both techniques have been reported to give significant WER reductions compared to ASR on single channel input, even if many distant-talking examples were used in training. What can be stated is

that the integrated approach requires more training data because it has to learn the multi-channel processing from the data. The approach with a separate beamformer in front of the acoustic model acts as a kind of regularizer, helping the overall system to settle on appropriate local minima of the networks, and thus requiring less training data and being computationally less demanding.

It should be mentioned that the integration can even be extended from subword unit DNN acoustic modeling to end-to-end speech recognition, which allows the beamforming components to be optimized jointly within the recognition architecture to improve the end-to-end speech recognition objective [37,38].

In [39], the effect of integrating enhancement and acoustic modeling was reported using a Google Home production system, where relative WER improvement of 8% to 28% was obtained by integrating WPE dereverberation and DNN based multichannel processing with the acoustic model of the production system.

C. Language modeling

Language modeling for the assistant is complex because of the ubiquity of applications it is used for. Taking example utterances from these interactions for the entire population of users allows us to estimate a LM that covers the domain as a whole. Language models used in the first pass are n-gram models predicting the next word and the sentence end based on a limited history of typically the three or four preceding words. Speech recognition systems often produce an n-best list in the first pass and apply a re-scoring second pass using log-linear or neural LM working on the complete sentence.

However, for an individual user, the actual entropy of the utterances they might utter is more restricted. For example, if users want to name a contact, they will likely pick a name that is in their contact list and less likely pick a name that is on the contact list of *any* user. In other words, a statically trained LM is a good fit to the domain, but has poor priors when it comes to an individual user. More general, the context in which an utterance is produced will have an impact on the content of the utterance. Digital assistant systems generally implement this adjustment by "biasing", i.e. adjusting the LM probabilities on-the-fly using the current context. The approach proposed in [40] achieves the biasing by boosting selected n-grams in the LM. An alternative approach used in [41] does an on-the-fly adjustment of the weights in a LM interpolation.

A second aspect resulting from the multitude of use cases is multi-linguality and code switching. Multi-lingual support for utterance-by-utterance language switching is implemented following the approach proposed in [42] by running several speech recognition systems, one per language, in

parallel. The best system and hence language is chosen after recognition is completed, either solely based on the scores of the language-dependent systems or supported by a language identification component. Code-switching within an utterance is usually implemented by adding a small degree of multi-linguality directly to the language model, e.g., an Indian-English speech recognition system usually covers also a limited set of common Hindi, Telugu, etc., phrases. A special case for a virtual assistant are catalogs, e.g., for supporting a music or shopping domain, where multi-lingual content is common. For example, users of an Indian-English system often ask for music or video titles in their native Indic language.

VI. Text-to-Speech Synthesis

Most of the smart loudspeakers have no screen to display information. On these devices, audio is the most natural way to provide responses to users, and text-to-speech (TTS) synthesis is used to generate spoken responses.

In the back-end of these digital home assistants, a natural language generation (NLG) module translates raw data into an understandable text in a markup language. A TTS system takes the markup text as its input and renders speech output. It consists of text analysis (front-end) and speech synthesis (back-end) parts. The text analysis part includes a series of natural language processing modules, such as sentence segmentation, word segmentation, part-of-speech tagging, and dictionary lookup and grapheme-to-phoneme pronunciation conversion. The speech synthesis part is usually a cascade of prosody prediction and waveform generation modules.

In the digital home assistant domain the text analysis part can access more contextual information than in other domains (e.g., synthesizing speech for a website), as the NLG module can provide it via markup language. For example, sometimes it is difficult to disambiguate pronunciation of a place name only from a written text. However, the NLG system can access its knowledge base to resolve the ambiguity and provide it via markup. Furthermore, the front-end can also incorporate explicit annotations providing hints about prosody and discourse domain [43]. Such coupling between NLG and TTS modules allows better synthesis in this domain.

In the back-end, either an example-based (concatenative) or a model-based (generative) approach is used in the waveform generation module. The former finds the best sequence of small waveform units (e.g., half-phone, phone, diphone-level) from a unit database given a target linguistic or acoustic features derived from an input text. The latter first learns a mapping function from a text to speech by a model, then predicts a speech waveform given a text and the trained model. The concatenative approach is known 1) to require a large amount of speech data from a

single speaker; 2) to have a large footprint; 3) to be computationally less expensive; 4) to have natural segmental quality but to sound discontinuous. On the other hand, the generative approach is known 1) to require less data or trainable from data from multiple speakers; 2) to have a small footprint; 3) to be computationally expensive; 4) to have smooth transitions but achieve relatively poor vocoder quality. Again, achieving low latency is critical for use in digital home assistants. Vendors choose different approaches to synthesize naturally sounding speech with low latency. We discuss two quite different solutions in the following to illustrate the range of options.

The Siri Team at Apple developed on-device deep learning-guided hybrid unit selection concatenative TTS system [43] to achieve these goals. Conventionally, hidden Markov models (HMMs) were used in hybrid unit selection TTS systems. Later, HMMs were replaced by deep and recurrent mixture density networks (MDNs) to compute probabilistic acoustic targets and concatenation costs. Multiple levels of optimizations (e.g., long units, preselection, unit pruning, local caching, parallel computation) enable the system to produce high quality speech with acceptable footprint and computational cost. Additionally, as an on-device system, it can synthesize speech without internet connection, allowing on-device low-latency streaming synthesis. Combined with a higher sampling rate (22 kHz $\rightarrow$ 48 kHz) and better audio compression, the system achieved significant improvements over their conventional system. This Siri deep learning-based voices have been used since iOS 10.

On the other hand, Google Home uses a server-side generative TTS system to achieve these goals. As Google's TTS system is running on servers, internet connection is essential. However, even on WiFi-connected smart loudspeakers, internet connection can be unstable. Audio streaming with unstable connection causes stuttering within a response. To prevent stuttering, no streaming is used in Google's TTS service; after synthesizing entire utterance, the server sends the audio to a device. The device starts playing the audio after receiving the entire response. Although this approach improves user experience, achieving low latency becomes hard. To achieve high-quality TTS with low latency, they developed the Parallel WaveNet-based TTS system [44]. Conventional generative TTS systems often synthesized "robotic" sounding vocoded speech. The introduction of sample-level auto-regressive audio generative models, such as WaveNet [45], has drastically improved the naturalness. However, it is computationally expensive and difficult to be parallelized due to its autoregressive nature. Parallel WaveNet introduced probability density distillation, which allows to train a parallel feed-forward network from an auto-regressive network, with no significant difference in segmental naturalness. Thanks to the parallelization friendly architecture

of Parallel WaveNet, by running it on tensor processing units (TPUs), it achieved 1,000 times speed up (20 times faster than real time) relative to the original auto-regressive WaveNet, while keeping its capability to synthesize high-fidelity speech samples. This Parallel WaveNet-based voices have been used in Google Assistant since October 2017.

VII. Fully hands-free interaction

Digital home assistants have a completely hands-free voice-controlled interface. This has important and challenging implications for the speech processing systems. The first, most obvious one is that the device has to be always listening to recognize if it is addressed by a user. But there are others as well, as will be explained in the following.

A. Wake-Up word detection

To detect if a user is addressing the device, a wake-up keyword, e.g., ”Alexa”, ”OK Google” or ”Hey Siri” is defined. If this word is detected, the device concludes that the following speech is meant for it. It is extremely important for user satisfaction that this keyword detection works very reliably, with both very low false alarm and high recall rates. This, however, is not easy, in face of the poor signal quality, see Fig. 2. Certainly, long wake-up words are easier to detect than short ones. However, because the keyword acts as the ”name” of the device, its choice is influenced by marketing aspects, leaving not so much room for engineering considerations. Another requirement is low latency. The system must answer as quickly as a human would do. Furthermore, one has to bear in mind that the keyword spotting algorithm runs on the device. This is different from the ASR component which is server-borne. Therefore, memory footprint and computational load considerations play also an important role [6].

In one approach that has been proposed in [46], a voice activity detection (VAD) is used in a first step to reduce computation, so that the search for a keyword is only conducted if speech has been detected. If speech is detected, a sliding window, whose size depends on the length of the keyword, is swept over the data, and a DNN classifier operates on the frames inside the window. In its simplest form classification is based upon a fully connected DNN, without any time alignment, resulting in significantly lower computational costs and latency compared to ASR. Then, max-pooling along the time axis is carried out on the DNN posteriors to arrive at a confidence score for the presence of a keyword.

To improve detection accuracy, convolutional [47], time delay [48], or recurrent network layers [49] have been proposed, as well as subword modeling of the keyword and the background speech

using a DNN-HMM architecture [50], all of which aiming at exploiting the temporal properties of the input signal for classification. To further reduce false alarm rates multi-stage keyword detection algorithms have been developed, where initial hypotheses are rechecked using cues like keyword duration, individual likelihoods of the phones comprising the keyword, etc. [50]. This second-stage classifier is again realized by a DNN. The experiments in [50] show that using subword based background models can reduce false accept rates (FARs) by about 37% relative at a fixed false rejection rate (FRR) of 4%. The work also shows the effectiveness of the two-stage approach which can reduce FARs by up to 67% relative at a 4% FRR. A different voice trigger detection system was proposed in [51] where robustness and computational efficiency are achieved using a two-pass architecture.

B. End-of-query detection

Not only the beginning of device-directed speech has to be detected. It has also to be determined quickly and accurately when the user has finished speaking to the system. However, speech pauses must not be taken falsely as the end of the query, nor must ambient noise, e.g., a TV running in the background, be taken as the continuation of an utterance.

From these considerations it is clear that a VAD can be no more than one source of information about the end of the user query [52]. Another source of information is the ASR decoder itself. Indeed, because the end-of-query detection is carried out on the server, the ASR engine is available for this task, and its acoustic and language model can be leveraged to identify the end of device-directed speech. An indication of this is if the active decoder hypotheses indicate end of sentence, followed by silence frames. Since low latency is important the decision cannot be postponed until all competing search hypotheses inside the ASR decoder have died out. To nevertheless achieve a high degree of reliability it has been proposed to average over all active search hypotheses with this property [53]. Yet another cue for the end of the user query is the recognized word sequence. Those sources of information, VAD, ASR decoder search properties and 1-best word/character hypothesis can be expressed as fixed-length features which are input to a dedicated end-of-query DNN classifier [54].

C. Second-turn device-directed speech classification

For a natural interaction, it is desirable that the system detects if another query is meant for it, without the user having to repeat the wake-up keyword again (Example: *"Hey Cortana, what*

is the weather today?" – System answer – *"and what about tomorrow?"*). One approach towards this functionality is to use a specific DNN classifier which rests its decisions on similar features as the end-of-query detector [55]: a fixed-length acoustic embedding of the utterance computed by a LSTM, the ASR decoder related features, e.g., the entropy of the forward probability distribution (large entropy indicating non-device-directed speech), and features related to the 1-best word/character sequence hypothesis. Those features are combined and input to a dedicated second-turn device-directed speech DNN detector.

An additional source of information to detect device-directed speech are the speaker characteristics, because the second turn is spoken by the same speaker as the first. Actually, all speech following the wake-up keyword and spoken by the same speaker can be considered to be device directed. Thus, a speaker embedding vector can be computed from the detected keyword speech. This embedding can be used to make an acoustic beamformer speaker dependent [56], and to improve the end-of-query and second-turn device-directed speech detection [57]. The encoder for mapping the keyword speech to an embedding is learned jointly with the classifier detecting device-directedness. Thus the classifier learns in a data-driven way what speaker and speech characteristics are relevant for detecting device-directedness.

D. Speaker identification

When digital home assistants are used by multiple members of a household, it is necessary to understand both what the user is asking for and who the user is. The latter is important to correctly answer queries like "When is my next appointment?". To do so, the system must perform utterance-by-utterance speaker identification. Speaker identification algorithms can be text-dependent, typically based on the wake-up keyword, [58,59] or text independent [60,61], and run locally on device or on the server. An enrollment process is typically necessary such that the assistant can associate speech with a user profile. Enrollment can be implemented by explicitly asking a user to provide an identity and a few example phrases. An alternative approach is for the assistant to identify speakers based on analyzing past utterances and next time when hearing a known speaker to ask for providing an identity.

VIII. CASE STUDY

To illustrate the impact of front-end multi-channel signal processing on ASR, the Engineering Team at Apple evaluated the performance of the far-field Siri speech processing system on a large speech test set recorded on HomePod in several acoustic conditions [6]:

- Music and podcast playback at different levels
- Continuous background noise, including babble and rain noise
- Directional noises generated by household appliances such as a vacuum cleaner, hairdryer, and microwave
- Interference from external competing sources of speech.

In these recordings, the locations of HomePod and the test subjects were varied to cover different use cases, for example, in living room or kitchen environments where HomePod was placed against the wall or in the middle of the room.

The performance of Siri online multi-channel signal processing was investigated in a real setup, where the trigger detection and subsequent voice command recognition jointly affect the user experience. Therefore, two objective Siri performance metrics, namely the false rejection rates (FRRs) and the word error rates (WERs), are reported.

Figure 5 shows the FRRs. The triggering threshold is the same in all conditions to keep the false alarm rates to a minimum. It can be observed that mask-based noise reduction is suitable in most acoustic conditions except for the multi-talker scenario, which is well handled by the stream selection system. For example, in the competing talker case, the absolute FRR improvement of the multi-stream system is 29.0% when compared to mask-based noise reduction, which has no source separation capability, and 30.3% when compared to the output of the baseline DSP system (that includes echo cancellation and dereverberation). The gap between mask-based noise reduction and the multi-stream system becomes smaller in other acoustic conditions. Overall, there is a clear trend of healthy voice trigger detection improvement when mask-based noise reduction and source separation techniques (stream selection) are used.

Figure 6 shows the WERs achieved by combining the multi-channel signal processing based on deep learning with the speech recognizer trained offline using internally-collected live data from HomePod to augment an existing training set, which was found to improve the ASR performance [6]. More details on data combination strategies to train acoustic models can be found in [2,3]. The blue portion of the bar represents the error rate of the triggered utterances, and the green portion represents the error rate due to falsely rejected utterances (missed utterances). Because triggered utterances can be different using one processing algorithm or another in different acoustic conditions, the WER numbers are directly influenced by the trigger performance. Different numbers of words are used for evaluation in the blue portion of the bars since the corresponding number of false rejections are significantly different for each case. It is obvious that the optimal

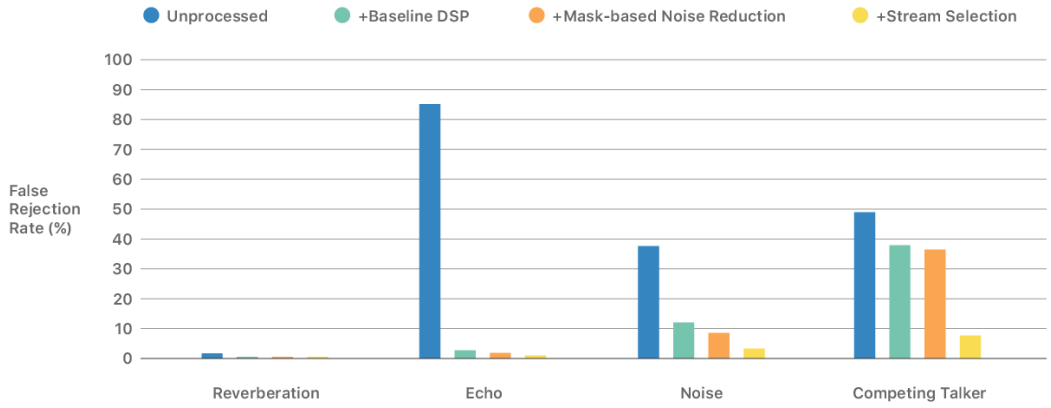

Fig. 5. False rejection rates of ”Hey Siri” detector in several acoustic conditions: reverberation, echo, noise and competing talker. ‘+ Baseline DSP’ refers to the baseline DSP. ‘+ Mask-based Noise Reduction’ refers to the baseline DSP and mask-based noise reduction. ‘+ Stream Selection’ refers to the baseline DSP, mask-based noise reduction, and stream selection [6].

and incremental integration of different speech processing technologies substantially improves the overall WERs across conditions [6]. More specifically, the WER relative improvements are about 40%, 90%, 74%, and 61% in the four investigated acoustic conditions of reverberant speech only, playback, loud background noise, and competing talker, respectively [6].

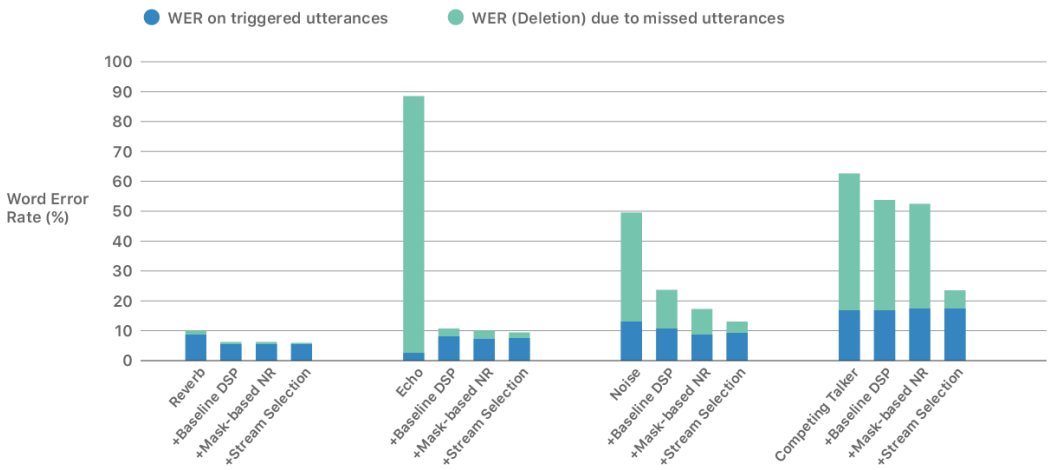

Fig. 6. Word error rates in several acoustic conditions (from left to right): reverberation, echo, noise and competing talker. ‘+ Baseline DSP’ refers to the baseline DSP that includes echo cancellation and dereveberation. ‘+ Mask-based NR’ refers to the baseline DSP and mask-based noise reduction. ‘+ Stream Selection’ refers to the baseline DSP, mask-based noise reduction, and stream selection [6].

IX. Summary and outlook

The article has given an overview of the speech processing challenges and solutions of digital home assistants. While deep learning is the method of choice to overcome many of these challenges, it is apparent that there is more to it than just training a deep neural black-box classifier on sufficiently large data sets. A clever interplay of signal processing and deep learning had to be developed to realize reliable far-field fully hands-free spoken interaction. The great success of this new class of products comes with new challenges, such as how to extend the range of applications and supported languages in an economically sensible way? Due to their conceptual simplicity, end-to-end ASR architectures appear to be one way to cope with those new challenges. But more research is needed until those new concepts have proven effective on the quite unique and demanding challenges of smart loudspeakers. For what is already possible today, you are invited to watch the promotional video by IEEE [62], illustrating that smart loudspeakers showcase signal processing at its best.

Authors

Reinhold Haeb-Umbach: Reinhold Haeb-Umbach (haeb@nt.uni-paderborn.de) is a professor of Communications Engineering at Paderborn University, Germany. He holds a Dr.-Ing. degree from RWTH Aachen University, and has a background in speech research both in an industrial and academic research environment. His main research interests are in the fields of statistical signal processing and pattern recognition, with applications to speech enhancement, acoustic beamforming and source separation, as well as automatic speech recognition and unsupervised learning from speech and audio. He (co-)authored more than 200 scientific publications, and recently co-authored the book Robust Automatic Speech Recognition – a Bridge to Practical Applications (Academic Press, 2015). He is a fellow of the International Speech Communication Association (ISCA).

URL: https://ei.uni-paderborn.de/nt/personal/arbeitsgruppe/mitarbeiter/haeb-umbach/

Shinji Watanabe: Shinji Watanabe (shinjiw@ieee.org) is an Associate Research Professor at Johns Hopkins University, Baltimore, MD, USA. He received his B.S., M.S., PhD (Dr. Eng.) degrees in 1999, 2001, and 2006, from Waseda University, Tokyo, Japan. He was a research scientist at NTT Communication Science Laboratories, Kyoto, Japan, from 2001 to 2011, a visiting scholar in Georgia institute of technology, Atlanta, GA in 2009, and a Senior Principal Research Scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA from

2012 to 2017. His research interests include automatic speech recognition, speech enhancement, spoken language understanding, and machine learning for speech and language processing. He has been published more than 150 papers and received several awards including the best paper award from the IEICE in 2003.

URL: https://sites.google.com/view/shinjiwatanabe

Tomohiro Nakatani: Tomohiro Nakatani (tnak@ieee.org) is a Senior Distinguished Researcher of NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan. He received the B.E., M.E., and PhD degrees from Kyoto University, Kyoto, Japan, in 1989, 1991, and 2002, respectively. He was a Visiting Scholar of the Georgia Institute of Technology for a year from 2005 and a Visiting Assistant Professor at Nagoya University from 2008 to 2017. He is currently a member of the IEEE SPS Speech and Language Processing Technical Committee. His research interests are in audio signal processing technologies for intelligent human-machine interfaces, including dereverberation, denoising, source separation, and robust ASR.

URL: http://www.kecl.ntt.co.jp/icl/signal/nakatani/

Michiel Bacchiani: Michiel Bacchiani (michiel@google.com) has been a speech researcher with Google for the past 14 years. He currently manages a research group in Google Tokyo focused on jointly modeling of speech and natural language understanding. Previously he managed the acoustic modeling team responsible for developing novel algorithms and training infrastructure for all speech recognition applications backing Google services. Before joining Google, Michiel Bacchiani worked as a member of technical staff at IBM Research, as a technical staff member at AT&T Labs Research and as a research associate at Advanced Telecommunications Research labs in Kyoto, Japan.

URL: https://research.google.com/pubs/MichielBacchiani.html

Björn Hoffmeister: Björn Hoffmeister (bjornh@a2z.com) is a Sr. Science Manager at Amazon leading Alexa speech, the automatic speech recognition R&D group that drives technology advances in ASR across all languages and provides ASR for all speech powered Amazon devices. As a founding member of Alexa research, he developed the wake word detection solution. With the Echo launch in 2014, Björn managed and grew the speech R&D group in the US, which supports speech for all Amazon. In 2015 Björn was leading the R&D efforts for the Alexa skills kit (ASK) project, which powers now 50,000+ third party skills for Alexa. In 2016 he helped launch the AWS Lex service based on the Alexa speech and skills technology.

URL: https://www.linkedin.com/in/bhoffmeister

Michael L. Seltzer: Michael L. Seltzer (mikeseltzer@fb.com) is a Research Scientist in the Applied Machine Learning division of Facebook. He received the Sc.B. degree with honors from Brown University and the M.S. and Ph.D. degrees from Carnegie Mellon University. From 1998 to 2003, he was a member of the Robust Speech Recognition Group, Carnegie Mellon University. From 2003–2017, he was a member of the Speech and Dialog Research Group at Microsoft Research. In 2006, he received the IEEE SPS Best Young Author award for his work optimizing microphone array processing for speech recognition. His current interests include speech recognition in adverse environments, acoustic modeling and adaptation, neural networks, microphone arrays, and machine learning for speech and audio applications.

URL: https://www.linkedin.com/in/michael-seltzer-a3815382/

Heiga Zen: Heiga Zen (heigazen@google.com) is a Senior Staff Research Scientist at Google. He received the A.E. degree from Suzuka National College of Technology, Suzuka, Japan, in 1999, and the Ph.D. degree from Nagoya Institute of Technology, Nagoya, Japan, in 2006. He was an Intern/Co-Op researcher at the IBM T.J. Watson Research Center, Yorktown Heights, NY (2004–2005), and a Research Engineer at Toshiba Research Europe Ltd. Cambridge Research Laboratory, Cambridge, UK (2008–2011). At Google, he was with the Speech team from July 2011 to July 2018, then joined the Brain team from August 2018.

URL: https://ai.google/research/people/HeigaZen

Mehrez Souden: Mehrez Souden (msouden@apple.com) is a Senior Audio and Speech Processing Engineer at the Interactive Media Group, Apple Inc. He received his Ph.D. and M.Sc. degrees from the Institut National de la Recherche Scientifique, University of Quebec, Montreal, QC, Canada, in 2010 and 2006, respectively. From 2010 to 2012, he was with the Nippon Telegraph and Telephone (NTT) Communication Science Laboratories, Kyoto, Japan. From 2013 to 2014, he was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, and from 2014 to 2015, he was with Intel Corporation. In 2016, he joined Apple Inc., working on signal processing and machine learning with emphasis on acoustics and speech. Mehrez Souden published more than fifty papers, and he was the recipient of the postdoctoral fellowship from the National Sciences and Engineering Research Council (NSERC) in 2013 and the Alexander-Graham-Bell Canada graduate scholarship from NSERC in 2008.

URL: https://www.linkedin.com/in/mehrez-souden-3695a29/

REFERENCES

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[2] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech*, 2018, pp. 1561–1565.

[3] K. Kinoshita, M. Delcroix, S. Gannot, E. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A. Sehr, and T. Yoshioka, "A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, 2016.

[4] M. Harper, "The automatic speech recogition in reverberant environments (ASpIRE) challenge," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2015, pp. 547–554.

[5] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, Nov 2012.

[6] Audio Software Engineering and Siri Speech Team. (2018, Dec) Optimizing Siri on HomePod in far-field settings. Apple Inc. [Online]. Available: https://machinelearning.apple.com/2018/12/03/optimizing-siri-on-homepod-in-far-field-settings.html

[7] J. Benesty, T. Gänsler, D. Morgan, M. Sondhi, and S. Gay, *Advances in network and acoustic echo cancellation*. Springer, 2001.

[8] B. Schwartz, S. Gannot, and E. A. P. Habets, "Online speech dereverberation using Kalman filter and EM algorithm," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 394–406, Feb 2015.

[9] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Blind speech dereverberation with multi-channel linear prediction based on short time fourier transform representation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[10] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2012.

[11] L. Drude, C. Boeddeker, J. Heymann, K. Kinoshita, M. Delcroix, T. Nakatani, and R. Haeb-Umbach, "Integration neural network based beamforming and weighted prediction error dereverberation," in *Proc. Interspeech*, September 2018.

[12] T. Yoshioka, H. Tachibana, T. Nakatani, and M. Miyoshi, "Adaptive dereverberation of speech signals with speaker-position change detection," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 3733–3736.

[13] J. Caroselli, I. Shafran, A. Narayanan, and R. Rose, "Adaptive multichannel dereverberation for automatic speech recognition," in *Proc. Interspeech*, 2017.

[14] Results of the 5th CHiME speech separation and recognition challenge. [Online]. Available: http://spandh.dcs.shef.ac.uk/chime_challenge/results.html

[15] C. Boeddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, "Exploring practical aspects of neural mask-based beamforming for far-field speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018.

[16] J. Heymann, M. Bacchiani, and T. Sainath, "Performance of mask based statistical beamforming in a smart home scenario," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[17] N. Ito, S. Araki, and T. Nakatani, "Complex angular central Gaussian mixture model for directional statistics in mask-based microphone array signal processing," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1153–1157.

[18] D. H. Tran Vu and R. Haeb-Umbach, "Blind speech separation employing directional statistics in an expectation maximization framework," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 241–244.

[19] N. Q. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.

[20] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[21] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[22] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[23] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks." in *Proc. Interspeech*, 2016, pp. 1981–1985.

[24] J. Heymann, L. Drude, and R. Haeb-Umbach, "A generic neural acoustic beamforming architecture for robust multi-channel speech processing," *Computer Speech & Language*, 2017.

[25] Y. Liu, A. Ganguly, K. Kamath, and T. Kristjansson, "Neural network based time-frequency masking and steering vector estimation for two-channel MVDR beamforming," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 6717–6721.

[26] M. Pedersen, J. Larsen, U. Kjems, and L. Parra, "A survey of convolutive blind source separation methods," *Multichannel Speech Processing Handbook*, pp. 114–126, Nov 2007.

[27] J. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: discriminative embeddings for segmentation and separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016.

[28] D. Yu, M. Kolbaek, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[30] J. B. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, Apr. 1979.

[31] E. Hadad, F. Heese, P. Vary, and S. Gannot, "Multichannel audio database in various acoustic environments," *Proc. International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 313–317, 2014.

[32] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home," in *Proc. Interspeech*, 2017, pp. 379–383.

[33] M. L. Seltzer, B. Raj, R. M. Stern *et al.*, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 489–498, 2004.

[34] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, May 2017.

[35] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5745–5749.

[36] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "BEAMNET: End-to-end training of a beamformer-supported multi-channel ASR system," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[37] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," *CoRR*, vol. abs/1712.01769, 2017. [Online]. Available: http://arxiv.org/abs/1712.01769

[38] T. Ochiai, S. Watanabe, T. Hori, and J. R. Hershey, "Multichannel end-to-end speech recognition," in *ICML*, 2017.

[39] B. Li, T. N. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Punduk, K. Chin, K. C. Sim, R. J. Weiss, K. W. Wilson, E. Variani, C. Kim, O. Siohan, M. Weintraub, E. McDermott, R. Rose, and M. Shannon, "Acoustic modeling for Google Home," in *Proc. Interspeech*, 2017.

[40] P. S. Aleksic, M. Ghodsi, A. H. Michaely, C. Allauzen, K. B. Hall, B. Roark, D. Rybach, and P. J. Moreno, "Bringing contextual information to Google speech recognition," in *Proc. Interspeech*, 2015.

[41] A. Raju, B. Hedayatnia, L. Liu, A. Gandhe, C. Khatri, A. Metallinou, A. Venkatesh, and A. Rastrow, "Contextual language model adaptation for conversational agents," in *Proc. Interspeech*, 2018.

[42] H. Lin, J. Huang, F. Beaufays, B. Strope, and Y. Sung, "Recognition of multilingual speech in mobile applications," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 4881–4884.

[43] T. Capes, P. Coles, A. Conkie, L. Golipour, A. Hadjitarkhani, Q. Hu, N. Huddleston, M. Hunt, J. Li, M. Neeracher, K. Prahallad, T. Raitio, R. Rasipuram, G. Townsend, B. Williamson, D. Winarsky, Z. Wu, and H. Zhang, "Siri on-device deep learning-guided unit selection text-to-speech system," in *Proc. Interspeech*, 2017, pp. 4011–4015.

[44] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, 2018, pp. 3918–3926.

[45] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[46] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 05 2014, pp. 4087–4091.

[47] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, 2015.

[48] K. Kumatani, S. Panchapagesan, M. Wu, M. Kim, N. Strom, G. Tiwari, and A. Mandai, "Direct modeling of raw audio with dnns for wake word detection," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 252–257.

[49] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Proc. of the 17th International Conference on Artificial Neural Networks*, ser. ICANN'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 220–229.

[50] M. Wu, S. Panchapagesan, M. Sun, J. Gu, R. Thomas, S. N. P. Vitaladevuni, B. Hoffmeister, and A. Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5494–5498.

[51] Siri Team. (2017, Oct) Optimizing Siri on HomePod in far-field settings. Apple Inc. [Online]. Available: https://machinelearning.apple.com/2017/10/01/hey-siri.html

[52] M. Shannon, G. Simko, S.-y. Chang, and C. Parada, "Improved end-of-query detection for streaming speech recognition," in *Proc. Interspeech*, 2017.

[53] B. Liu, B. Hoffmeister, and A. Rastrow, "Accurate endpointing with expected pause duration," in *Proc. Interspeech*, 2015.

[54] 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 *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[55] S. Mallidi, R. Maas, K. Goehner, R. A., S. Matsoukas, and B. Hoffmeinster, "Device-directed utterance detection," in *Proc. Interspeech*, 2018.

[56] J. Wang, J. Chen, D. Su, L. Chen, M. Yu, Y. Qian, and D. Yu, "Deep extractor network for target speaker recovery from single channel speech mixtures," in *Proc. Interspeech*, 2018.

[57] R. Maas, S. H. K. Parthasarathi, B. King, R. Huang, and B. Hoffmeister, "Anchored speech detection," in *Proc. Interspeech*, 2016.

[58] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[59] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2016.

[60] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Interspeech*, 2017.

[61] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[62] Signal processing in home assistants. IEEE. [Online]. Available: https://www.youtube.com/watch?v=LJ54btWttdo