Exploring the application of synthetic audio in training keyword spotters

Andrew Werchniak[1], *Roberto Barra Chicote*[1], *Yuriy Mischenko*[1], *Jasha Droppo*[1], *Jeff Condal*[1], *Peng Liu*[1], *Anish Shah*[1]

[1]Alexa Speech, Amazon.com

{wercha, rchicote, yuriym, drojasha, jccondal, liupng, anishsh}@amazon.com

Abstract

The study of keyword spotting, a subfield within the broader field of speech recognition that centers around identifying individual keywords in speech audio, has gained particular importance in recent years with the rise of personal voice assistants such as Alexa. As voice assistants aim to rapidly expand to support new languages, keywords, and use cases, stakeholders face the issue of limited training data for these unseen scenarios. This paper details some initial exploration into the application of Text-To-Speech (TTS) audio as a "helper" tool for training keyword spotters in these low-resource scenarios. In the experiments studied in this paper, the careful mixing of TTS audio with human speech audio during training led to a reduction of over 11% in the detection-error-tradeoff (DET) area under the curve (AUC) metric.

Index Terms: keyword spotting, speech recognition, data augmentation, speech synthesis

1. Introduction

Over the past few years, voice assistants such as Amazon's Alexa, Google Assistant, and Apple's Siri have risen rapidly in popularity, to the point that they have become a staple of everyday life for many people across the globe. Alexa, in particular, now has tens of millions of users who interact with their devices in myriad different languages and accents [1]. One of the most important steps in ensuring a frictionless experience for customers of voice assistants is "waking" the device up for interaction when the customer intends to use the device, and, importantly, not "waking" when the customer does not. This is accomplished with specialized "wakeword models" that detect keywords spoken by users and initiate interactions with the device [2, 3, 4].

The expansion of an existing assistant to new languages, wakewords, settings, and applications presents a challenging data problem. Compared to the flagship wakeword model, there may be very little training data available. And, the methods used to bootstrap the flagship wakeword model are too costly and time consuming to scale quickly and efficiently.

This paper explores addressing this data problem via amending existing low-resource wakeword data preparation techniques using synthetic training data created with automatic text-to-speech (TTS) systems, specifically Amazon Polly and an internal TTS model trained support over 1M unique speaker profiles. It is shown that in the application of wakeword model training, accuracy improvements of up to 11% (in terms of Area Under the Curve) can be gained by including "synthetic" voice data at the right weighting with "organic" voice data during training. The rest of the paper is as follows: Section 2 discusses past work studying the application of TTS audio in speech recognition systems; Section 3 details the methodology, including the data preparation, model training, and model evaluation; Section 4 details the experimental results; and Section 5 summarizes the conclusions and future work to build on the results.

2. Related Work

Some previous research has been dedicated to the application of synthetic audio in training automatic speech recognition (ASR) systems. Large vocabulary ASR models of architectures varying from Gaussian Mixture Models (GMM)/Hidden Markov Models (HMM) [5] to Convolutional Neural Network(CNN)/Connectionist Temporal Classification (CTC) models [6] to more modern attention-based acoustic-to-word models [7, 8] have all been shown to benefit from the addition of TTS data at varying levels and stages. However, it is worth noting that there may be limits to these benefits, as it has been shown that bispectral analysis can still differentiate with confidence between audio generated with state-of-the-art TTS systems and human audio[9], indicating that a mismatch may still exist between synthetic training audio and organic evaluation audio.

Regardless, the application of synthetic data in training low-resource keyword spotter systems has shown promise in recent experiments. Specifically, it was demonstrated that by utilizing a pre-trained speech-embedding model with approximately 400K parameters and weights initialized using human audio data, subsequent training on approximately 2000 synthetic voice examples produced a model with performance only slightly worse than the same model trained with the same number of organic audio examples [4]. The experiments in this paper build on these past works by applying the idea to a new model architecture and training environment. To the authors' knowledge, it is the first study to demonstrate clear accuracy improvements in a state-of-the-art keyword spotting system from the inclusion of TTS audio during model training.

3. Methodology

3.1. Model Architecture & Training Parameters

The experiments in this paper use small footprint word-level models (around 50K parameters), a model suitable for settings in which CPU and memory are extremely limited. Although it should be possible to train larger models in the same way, we focus on these smaller models so that the lack of diversity in the current TTS output is not a confounding factor in this pilot study.

The acoustic features used as input to the models are 20-dimensional log-filterbank energies (LFBEs) produced from a sliding window of 25ms computed every 10ms. The input segments span 80 frames (800 ms) with the target word aligned at the end of the segment to reduce online detection latency. All

models studied in this paper are of the same architecture, as detailed in Figure 1: a fully-connected DNN with batch normalization applied during training, ReLU activation functions on the hidden layers, and a binary SoftMax output layer. The models were trained using mini-batch gradient descent for 200K steps, employing the Adam optimizer with an initial learning rate of 0.0015, a batch size of 500 examples, and a cross-entropy loss function. For wakeword detection, the DNN posteriors are smoothed across the time dimension using a running average sliding window of 25 frames. In Alexa wakeword model training settings, these models provide a strong baseline performance of small footprint keyword spotting [10].

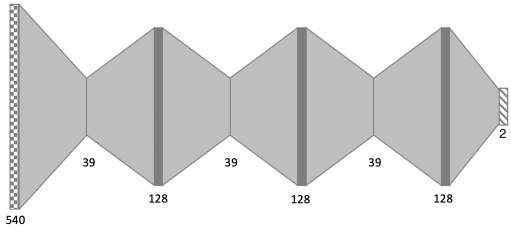

Figure 1: *50K Small Footprint Word-Level DNN Architecture. An input context of 80 temporal frames (downsampled 1-3) with 20 frequency dimensions is flattened into a 540-dimensional vector. Hidden layers alternate in size between 39 and 128 nodes. Output is a binary "keyword" or "not keyword" Softmax prediction.*

3.2. Baseline Training Dataset Preparation

Although the purpose of this paper is to measure the potential benefits of using TTS audio to improve the quality of models for low-resource training scenarios, the well-studied wakeword of "Alexa" for United States English was selected for experimentation due to the availability of quality annotated evaluation data.

The training data was selected as if "Alexa" were any other low-resource wakeword. To prepare the positive examples, six months' worth of production customer audio was transcribed using an offline automatic speech recognition (ASR) model. This corpus was filtered to remove all utterances where the initiating wakeword was "Alexa", and further filtered to only include utterances that contained "Alexa" later in the transcription. These examples typically represent speech not intended to wake the device, such as mentioned in passing or in conversational context; unlike "Alexa" used explicitly as a wakeword, such mentions may differ in intonation, rate of speech, and other patterns. This technique closely mimics the process used in development of models for new wakewords, for which we expect to only have in-passing mentions and not examples explicitly intended to wake the device. For the negative examples, a single day's worth of production traffic was transcribed using the same ASR model, and filtered such that none of the transcriptions contained the word "Alexa". The resulting dataset counts, which we call "organic" to differentiate from any synthetic data, are summarized in Table 1.

Table 1: *Baseline Training Dataset Counts*

Dataset	Positive Examples	Negative Examples
Organic	730127	718598

3.3. Amazon Polly Training Dataset Preparation

Amazon Polly was used as the TTS system for the first round of experiments studied in this paper. Polly is a cloud service available to AWS customers that converts text into lifelike speech. For the U.S. English locale, Polly supports 8 distinct voice profiles which can be further varied by including special instructions in the input utterance text [11].

To prepare positive examples, anonymized transcriptions of the top 625 utterances spoken over the past year were used as input text. The utterances were then each prepended with the "Alexa" wakeword, with 50% of them chosen at random also including a randomly selected "supported wakeword prefix" from the list ("hey," "hi," "okay," "yo") beforehand. These 625 utterance texts were then randomly augmented 10X by inserting Polly instructions in the text to (1) vary along the volume, timbre, speaking rate, and pitch prosody dimensions; (2) insert random human breath noises and pauses; and (3) randomly select certain utterances to be either whispered or spoken "softly." All 8 U.S. English Polly voices were used to synthesize one example of each augmented utterance text, yielding 50K near-field speech examples.

To prepare negative examples, anonymized transcriptions of an additional 625 common utterances from the past year were used. For 50% of the utterances chosen at random, a "close variant" of the wakeword was randomly selected to initiate the stream; these are words whose pronunciation sounds sufficiently close to that of the wakeword and might confuse a model. The close variants to "Alexa" used for this experiment were ("exclamation," "congresswoman," "Kevins car," "election," "Alaska," "I like the," "unacceptable," "I think so"). The other 50% were synthesized as generic speech examples with no initiator. These were then similarly augmented 10X using the same Polly options and synthesized using all 8 United Stated English Polly voices, for 50K audio examples.

Finally, using the same techniques described in [12], these datasets were merged and augmented 5X by randomly mixing in different room impulse responses to simulate far-field speech in different acoustic environments, and then further augmented 2X by randomly mixing common background noises. These steps were performed to increase the size of the training dataset as well as to more closely mimic the far-field speech audio expected by smart speaker devices. The resulting dataset counts are summarized in Table 2.

Table 2: *Amazon Polly Training Dataset Counts*

Dataset	Positive Examples	Negative Examples
Clean Near Field	50000	50000
Noisy Near Field	50000	50000
Clean Far Field	200000	200000
Noisy Far Field	200000	200000
Total	500000	500000

3.4. Novel TTS Training Dataset Preparation

Amazon Polly offers voices with human-like audio quality, but unfortunately it only offers 8 different voices for the United States English locale, and there are only so many ways that each voice can pronounce "Alexa." In order to experiment with a TTS system that offers greater speaker diversity, the Alexa Research team developed a neural TTS model able to synthesize speech from over 1 million distinct speaker profiles.

The system, detailed in Figure 2, consists of two modules: a context generation module and a neural vocoder module. The context generation module is an attention-based sequence-to-sequence network [13, 14] that predicts a Mel-spectogram given an input text. It has three preprocessing submodules: (1) a grapheme-to-phoneme encoder that converts the sequence of words into a sequence of phonemes plus augmented features such as punctuation marks and prosody-related features derived from the text (e.g. lexical stress); (2) a reference encoder that replicates the spectrogram used as the target in the optimization function as an additional input condition; and (3) a "voice profile" embedding input, generated by a speaker verification system similar to the one described in [15]. During inference, the neural network broadcasts and stacks the reference spectrogram and voice profile embedding with the sequence produced by the phonetic encoder and infers the Mel-spectrogram output sequence. Finally, the neural vocoder module, consisting of the architecture introduced in [16] and pretrained with 2000 utterances per each of the 74 voices from a proprietary database of paid voice actors, synthesizes speech audio out of the Mel-spectrograms generated by the first module.

The speaker verification system used to produce the voice profile embeddings for the context generation module was trained by sampling data from 20+K speakers from the 25+ languages available in the Mozilla Common Voice corpus [17]. The resulting learned dense space was leveraged to create over 1M synthetic voice profiles by linearly interpolating between voices from a set of 1K speakers enrolled in training the context generation module.

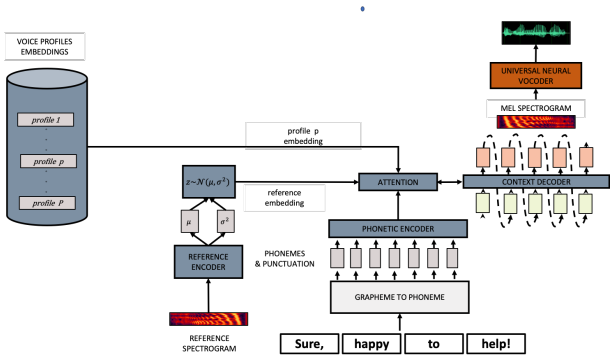

Figure 2: *Novel TTS System Architecture. The system consists of two modules. First, a context generation module predicts a Mel-spectrogram from input text, a reference spectrogram, and a speaker embedding. Second, a neural vocoder module produces a speech waveform from that Mel-spectrogram.*

A random sample of 90K of the voice profiles available in this model was selected to synthesize speech. For each voice profile, one positive "Alexa" example and one negative speech example were prepared. This dataset was then augmented in the same fashion as the Amazon Polly dataset, with noise and reverberation mixing. Finally, the training features were prepared by applying a 2-stage DNN/HMM-DNN "Alexa" teacher model; this served both to identify the exact location of the keyword in each training example and to filter out utterances for which the experimental TTS model may have produced nonconvergent results (note that this was a temporary solution; in future work we will know the exact location of the wakeword within the synthesized audio, and we will have greater confidence in the output of the TTS model, so filtering will be unnecessary). The resulting dataset counts are summarized in Table 3.

Table 3: *Novel TTS Training Dataset Counts*

Dataset	Positive Examples	Negative Examples
Clean Near Field	88187	82455
Noisy Near Field	88187	82455
Clean Far Field	353696	327994
Noisy Far Field	352696	327994
Total	882766	820898

4. Experimental Results

4.1. Evaluation Datasets

Two evaluation datasets using annotated audio were prepared. The first dataset represents standard Alexa interactions, while the second dataset represents trickier wakeword conditions (e.g. high noise environments, etc.). Although the first dataset contains orders of magnitude more data, it suffers from a strong selection bias. Thus, the two datasets are weighted 50/50 during evaluation in order to give fair evaluation of trickier conditions and better estimate online performance. The datasets are summarized in Table 4.

Table 4: *Evaluation Dataset Counts.*

Dataset	Count	Weighting
Standard Alexa Utterarnces	1,128,814	50%
Tricky Alexa Utterances	34,609	50%
Total	1,163,423	100%

4.2. Models Trained with Polly

The "organic" dataset, mined from customer audio, and "synthetic" dataset, produced using Polly, were combined to train five different models. In each case, all training examples were used and the datasets were given different weights when computing training loss. The paired Polly and organic weights were (0.0, 1.0), (0.25, 0.75), (0.50, 0.50), (0.75, 0.25) and (1.0, 0.0). The models were evaluated using end-to-end decoding on the evaluation datasets detailed in Table 4. The output was evaluated against the human ground truth labels, and performance was plotted on Detection-Error-Tradeoff (DET) curves, mapping false detection rate (1-precision) vs. miss rate (1-recall). Performance is reported in Table 5 as relative improvement in the Area Under the Curve (AUC) metric with respect to the purely organic baseline model. The model that uses a 0.75 organic, 0.25 Polly, weighting achieves a 2.62% improvement in AUC. Notably, these gains are sensitive to the chosen weights, as all other mixtures of Polly audio explored in training degraded end-to-end model performance. This indicates that at least within the scope of using Polly TTS data for training, organic speech audio provides important context towards mimicking the evaluation dataset distribution.

4.3. Models Trained with Novel TTS

Similarly, five different models were trained on various weightings of the "organic" dataset of mined customer audio and the "synthetic" dataset generated using our novel TTS system detailed above. The training loss weights, the decoding, and the performance evaluation was done as in the previous section.

Table 5: *Polly Models Evaluation Summary*

Training Dataset Weight		% AUC
Polly	Organic	Reduction
1.00	0.00	-75.6
0.75	0.25	-11.9
0.50	0.50	-10.3
0.25	0.75	2.62
0.00	1.00	0.00 (baseline)

The relative AUC reduction with repect to the baseline is summarized in Table 6. Significant accuracy improvements were achieved with the addition of synthetic audio data generated with the TTS system. The best results weight 0.75 synthetic data and 0.25 organic data, with a reduction in AUC by 11.3%. Compared with the models trained on Polly audio, it appears that this system is less sensitive to dataset weightings, as all models trained on a mixture of synthetic and organic audio outperform the baseline. However, it's clear that some human data still provides the model with important context, as the curve trained on 100% synthetic data attains an AUC value 28.9% higher than the baseline.

Table 6: *Novel TTS Models Evaluation Summary*

Training Dataset Weight		% AUC
Novel TTS	Organic	Reduction
1.00	0.00	-28.9
0.75	0.25	11.3
0.50	0.50	3.07
0.25	0.75	5.75
0.00	1.00	0.00 (baseline)

5. Conclusions

This paper improved on previous studies by demonstrating that when we lack sufficient human audio training data, TTS audio can provide valuable training data that improves model performance. Using TTS for wakeword model training is an important direction as could it potentially enable building models for new wakewords and/or locales with reduced or zero involvement of organic speech data. Our novel TTS system, trained to maximize the diversity in speaker profiles, generated data that, when combined with organic audio, showed improvements from the baseline across the board, up to a greater than 11% decrease in evaluation AUC. Similarly, it was demonstrated that it is possible to derive gains from data generated by Amazon Polly up to a few percent, if included at the correct weight with organic audio in the training datasets.

Clear limitations still remain with regards to the application of synthetic TTS audio towards keyword spotter model training. The most obvious limitations boil down to discrepancies in speaker quality and diversity compared with the organic audio typically used for training wakeword models at Alexa. Although Amazon Polly offers high-quality, human-sounding voices, it suffers from a dearth of speaker diversity, as only 8 voices are available for the U.S. English locale, compared with hundreds of thousands of speakers used to create the baseline dataset. Although our novel TTS architecture offers a number of individual voice profiles comparable (or even exceeding) those available from production data, the "quality" (measured qualitatively via listening exercise) of its synthesized audio is not yet on par with organic audio. Regardless of these limitations, the results detailed in this paper are an important first step in studying the potential application of TTS audio in training keyword spotters, and offer promising prospects to be improved upon with further research.

6. References

[1] "Alexa became even more natural and useful for customers in 2019." [Online]. Available: https://blog.aboutamazon.com/devices/alexa-became-even-more-natural-and-useful-for-customers-in-2019

[2] A. Raju, S. Panchapagesan, X. Liu, A. Mandal, and N. Strom, "Data augmentation for robust keyword spotting under playback interference," 2018. [Online]. Available: https://arxiv.org/pdf/1808.00563.pdf

[3] S. Sigtia, R. Haynes, H. Richards, E. Marchi, and J. Bridle.

[4] J. Lin, K. Kilgour, D. Roblek, and M. Sharif, "Training keyword spotters with limited and synthesized speech data," 2020. [Online]. Available: https://arxiv.org/pdf/2002.01322.pdf

[5] L. V. Rygaard, "Using synthesized speech to improve speech recognition for lowresource languages," 2015. [Online]. Available: https://parasol.tamu.edu/dreu2015/Rygaard/report.pdf

[6] J. Li, R. Gadde, B. Ginsburg, and V. Lavrukhin, "Training neural speech recognition systems with synthetic speech augmentation," 2018. [Online]. Available: https://arxiv.org/pdf/1811.00707.pdf

[7] M. Mimura, S. Ueno, H. Inaguma, S. Sakai, and T. Kawahara, "Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 477–484, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8639589

[8] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, "Speech recognition with augmented synthesized speech," 2019. [Online]. Available: https://arxiv.org/pdf/1909.11699.pdf

[9] E. A. AlBadawy, S. Lyu, and H. Farid, "Detecting ai-synthesized speech using bispectral analysis," *Workshop on Media Forensics at CVPR*, 2019. [Online]. Available: https://farid.berkeley.edu/downloads/publications/cvpr19/cvpr19b.pdf

[10] Y. Mishchenko, Y. Goren, M. Sun, C. Beauchene, S. Matsoukas, O. Rybakov, and S. N. P. Vitaladevuni, "Low-bit quantization and quantization-aware training for small-footprint keyword spotting," *IEEE ICMLA 2019*, 2019.

[11] [Online]. Available: https://docs.aws.amazon.com/polly/latest/dg/what-is.html

[12] Y. Gao, Y. Mishchenko, A. Shah, S. Matsoukas, and S. N. P. Vitaladevuni, "Modeling wake word spotters using limited data," *Interspeech 2019*, 2019.

[13] N. Prateek, M. Lajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood, "In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data," in *Proc. NAACL*, 2019, pp. 205–213.

[14] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and K. Viacheslav, "Effect of data reduction on sequence-to-sequence neural TTS," in *Proc. ICASSP*, 2019, pp. 7075–7079.

[15] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," 2017. [Online]. Available: https://arxiv.org/pdf/1705.02304.pdf

[16] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards Achieving Robust Universal Neural Vocoding," in *Proc. Interspeech*, 2019, pp. 181–185. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1424

[17] [Online]. Available: https://voice.mozilla.org/en/datasets