MONOPHONE-BASED BACKGROUND MODELING
FOR TWO-STAGE ON-DEVICE WAKE WORD DETECTION

Minhua Wu1, Sankaran Panchapagesan2∗, Ming Sun1, Jiacheng Gu1
Ryan Thomas3, Shiv Naga Prasad Vitaladevuni3, Bjorn Hoffmeister1, Arindam Mandal1

1Amazon.com, Inc. USA 2Google LLC. USA
wuminhua@amazon.com, panchi@gmail.com, mingsun@amazon.com

ABSTRACT
Accurate on-device wake word detection is crucial to products with far-field voice control such as the Amazon Echo. It is quite challenging to build a wake word system with both low False Reject Rate (FRR) and low False Alarm Rate (FAR) in real scenarios where there are various types of background speech, music or noise, especially when computational resources on the device is limited. In this paper, we introduce a two-stage wake word system based on Deep Neural Network (DNN) acoustic modeling, propose a new way to model the non-keyword background events using monophone-based units and present how richer information can be extracted from those monophone units for final wake word detection. Under the new system, we could get around 16% relative reduction in FRR when fixing the false alarm level, and about 37% relative reduction in FAR on the other hand if we maintain the miss rate. For the 2nd stage classifier itself, it is able to reduce the false alarm rate relatively by about 67% on top of 1st stage hypothesis with very few computational resources.

Index Terms— wake word detection, deep neural network, monophone-based units

1. INTRODUCTION
The Amazon Echo is a smart home device using voice-based user interface. With the Keyword Spotting (KWS) system running on the device, it is always listening for the wake word, such as "Alexa". Once the wake word is detected, it can stream audio to cloud and interpret voice commands. For this application, wake word detection is the first important step before interactions through distant speech recognition [1] and accurate on-device detection is crucially important for a great customer experience.

There is a rich literature on the topic of keyword spotting in continuous speech. In much of recent work, latency and computation are not concerns. Offline Large Vocabulary Continuous Speech Recognition (LVCSR) systems can be used to decode the audio and create transcripts or lattices, which can then be used for detecting the keyword(s) of interest [2] [3] [4] [5]. For online low-latency and computation-constrained KWS systems, Hidden Markov Models (HMM) were commonly used [6] [7] [8]. With the growing success of deep learning in recent years, Gaussian Mixture Models (GMM) for acoustic modeling under the HMM framework has been replaced with Deep Neural Network (DNN) in the field of Automatic Speech Recognition (ASR) [9]. This DNN-HMM hybrid approach using various neural network architecture has also been effectively applied for the wake word application [10][11][12][13][14]. Alternatively, some recent work proposed to build systems based on a single DNN or convolutional neural network (CNN) with no HMM involved [15] [16] [17]. RNN/LSTM based keyword spotting system has been proposed which can leverage longer temporal context [18] [19] [20] [21]. An end-to-end trained sequence-to-sequence model was also proposed recently [22]. There is no solid conclusion on whether the HMM-based KWS system is better than the non-HMM based system for real-time single wake word detection. Non-HMM based KWS system may have better discriminative capabilities, but it usually requires carefully designed post-smoothing algorithm to get the final decision. In this paper, we will focus on single wake word detection on resource-constrained embedded devices using HMM-based technique.

The wake word detector for experimentation in this paper has been designed to use a DNN-HMM decoder at first, followed by a light second stage classifier to make the final decision [23]. The small second stage classifier is meant to improve the discriminative power of the HMM-based KWS system without introducing too much computation and latency. The original system uses simple speech and non-speech events for background modeling. To better distinguish between the wake word and confusable background audio segments, we propose to incorporate monophone-based units to model the non-keyword background, and use those monophone statistics for further classification.

The paper is organized as follows. In section 2, we will briefly describe how the original two-stage wake word system works. In the next section, we will illustrate how we...
2. BASELINE TWO-STAGE WAKE WORD SYSTEM

2.1. 1st Stage DNN-HMM Decoder

An example 1st stage HMM decoding graph for the keyword "Alexa" is shown in Figure 1. Note that single-state HMMs for the phones are shown for simplicity. The foreground HMM consists of wake word phones and several non-speech frames at the beginning, while the background HMM consists of a loop over single-state speech and non-speech events. Viterbi decoding is performed on the HMM decoding graph using frames of acoustic features computed from the audio signal, and we are using a Deep Neural Network (DNN) based acoustic model to compute posteriors of different HMM states. A wake word is hypothesized by the first stage if the final state of the foreground HMM is reached and the difference between foreground and background log-likelihoods during the candidate segment exceeds a threshold. Various transition and exit penalties in both the foreground and background HMMs can be tuned for better accuracy, and a first stage DET curve can thus be obtained by plotting the lowest achievable FRR at a given FAR.

2.2. 2nd Stage Classifier

When the first stage log-likelihood ratio exceeds the threshold, the corresponding audio segment \( X \) which runs through is treated as a candidate wake word. Though candidates triggered from the first stage will be of different durations, they will be transformed to a feature vector \( v \) of dimension \( D (D = 67) \) based on how the candidate segment is aligned with the foreground HMM and what are the likelihoods for frames in the candidate given different states. Specifically, the feature vector \( v \) for the candidate wake word \( X \) includes information from both the entire segment and individual phone segments. Segment-level features include the duration, keyword likelihood score, normalized likelihood score and posterior for the keyword. For the phone-level features, we consider absolute and relative phone duration, phone log-likelihood, averaged phone/speech confidence scores, local context features such as left/central/right phone confidence, and entropy based features according to context phone confidence scores. A second stage classifier is then trained based on feature vectors extracted from wake word candidates to make the final decision. We use a small feed-forward neural network for experimentation here. For the consideration of detection latency, we are not using RNN based architecture or attention mechanism to summarize the information, but it could be tried in the future.

3. NEW WAKE WORD SYSTEM USING MONOPHONE-BASED BACKGROUND MODELING

3.1. 1st Stage DNN-HMM Decoder with Monophone-based Background Model

Instead of simply using speech and non-speech background events, we expand them to various monophones. With these new units introduced, our new background HMM becomes a phone-level unigram FST. Figure 2 is a simplified version of the new background model, since we are actually using 3-state HMM topology for these background monophones. We should still be able to use Viterbi decoding and DNN based acoustic model on the new decoding graph, but output targets of the DNN will be expanded accordingly.

3.2. New Feature Engineering for 2nd Stage Classifier

Since we now have a first stage wake word detector with more HMM states, we can extract richer information from them for better second verification. The original 67 feature dimensions illustrated in section 2.2 are still valid. Additionally, we come up with a new score \( \text{MatchScore}_{p,q} \) measuring the degree of match between each candidate's wake word phone segment \( p \) and every background monophone \( q \), as indicated in equation 1. For each frame \( X_t \) within one wake word phone \( p \), we take the maximum log likelihood among the three states of each background monophone \( q \), and average these log likelihoods over the phone duration of \( p \). We need to compute the match score for each wake word phone \( p \) with respect to every background monophone \( q \). Since the background model is only learned from non-keyword audio, we would expect lower
match score between the real wake word phone segment and background monophones, but higher match score between phone segment in false wake word hypothesis and background monophones. In this case, we could be able to distinguish better between real wake words and confusable segments among first stage candidates by introducing these new features. We are using $N_q = 44$ monophone units in the background model, so we could introduce 44 extra features for each wake word phone segment. Therefore, the new second stage classifier will end up with using a new feature vector of dimension 375 ($67 \times 44$). We could do feature selection to make the model work more efficiently, but this topic is not discussed under the scope of this paper.

$$\text{MatchScore}_{p,q} = \frac{1}{\text{Dur}_p} \sum_{t \in \text{Dur}_p} \max\{\log(P(X_t|Q^p_{t}\theta_{\text{BC}})), \log(P(X_t|Q^q_{t}\theta_{\text{BC}})), \log(P(X_t|Q^R_{t}\theta_{\text{BC}}))\}$$

$p \in$ wake word phones: 
\{SIL\text{Preceding}, AX_{\text{A}lexa}, L_{\text{A}lexa}, E\text{H}_{\text{A}lexa}, K_{\text{A}lexa}, S_{\text{A}lexa}, AX_{\text{E}deliver} \}
$q \in$ background monophones

4. EXPERIMENT RESULTS

4.1. Baseline Setup

The training data used in this work consists of several thousand hours of the real far-field data captured in various rooms. It contains approximately several hundred thousand subjects. Our development and test set contain tens of thousands of speech streams uttered by hundreds of subjects. Both the development and test set have approximately 30,000 wake word instances. We use a feed-forward DNN acoustic model in the first stage. The decoding graph consists of a foreground wake word HMM and a background speech/non-speech loop HMM. The input acoustic features to the DNN consists of 31 stacked frames of 20-dimensional Log mel-Filter-Bank Energies (LFBE) features (20 frames for left context and 10 frames for right context). The baseline DNN has 50 output targets corresponding to phoneme state under wake words, commands (e.g. “STOP”) and speech/non-speech events. The DNN acoustic model is at the same time trained to predict the Large Vocabulary Continuous Speech Recognition (LVCSR) targets, since this is known to improve the accuracy for wake word detection [10]. A weighted cross-entropy objective function is used where the loss stream corresponding to the LVCSR task is weighted by 0.1 while the loss stream corresponding to the wake word task is weighted by 0.9. Each wake word candidate segment hypothesized from the first stage is then transformed to a fixed 67-dimensional feature vector for final detection with the second stage classifier using a small feedforward neural network (NN). The feed-forward DNN acoustic model is chosen to have 4 hidden layers. Two sizes for the hidden layer are tried (896, 1024) respectively in the experiments. The GPU-based distributed DNN trainer described in [24] is utilized and the DNN is discriminatively pre-trained beforehand in a layer-wise manner [25] on a small subset of the training data.

4.2. Monophone-based System Setup

For the decoding graph in the new wake word system using monophone-based background, the foreground wake word HMM keeps the same, but the background HMM changes to be a phone-level unigram FST as plotted in figure 2 from section 3.1. Input features to the DNN acoustic model are still the same but wake word targets at the output layer are expanded. Non-keyword frames in the training data are now assigned with specific monophone states instead of either speech or non-speech label. The new DNN now has 178 output nodes including both wake word phone states and monophone states for the wake word task. Each keyword candidate segment is then transformed to a fixed 375-dimensional feature vector for the 2nd stage NN classifier.

4.3. Experiment Results

4.3.1. 1st Stage HMM Tuning

We first tune the two systems by varying penalties on the decoding graph. It turns out performance of the two systems are almost the same at this stage on the dev set. This is expected since both the baseline and the new system model foreground and background phones, and the new system is not using higher order language model but phone-level unigram. However, we can extract richer information from the new system’s first stage decoding results for further classification. For both the baseline and the new wake word system, we choose a first stage operating point under the similar level of recall at around 0.02 and proceed to build the second stage classifier.

4.3.2. End-to-end Evaluation

DET curves comparing the end-to-end performance of the two systems are displayed in figure 3. We tried several sizes for hidden layers of the 2nd stage NN classifier with two hidden layers. These curves are obtained by sweeping the final accept threshold in each system. The dashed curve corresponds to the overall performance of the baseline wake word system, and the solid curve corresponds to the overall performance of the new wake word system using monophone-based background model. We can see the new system performs generally better than the baseline, which indicates the additional features derived from the new wake word system are effectively helpful. When we vary the size for hidden layers of the 2nd stage NN classifier (from figure 4a to 3d), we can see some gain when changing
Figure 4 compares performance with and without the 2nd stage classifier on both the baseline and the new system. It turns out that if we fix the false reject rate to be at 0.04, we could reduce the FAR by about 45% relatively by deploying the 2nd stage classifier for the 4×896 speech/non-speech background based wake word system. The 2nd stage classifier seems to be more effective for the monophone background based wake word system. We can achieve about 67% relative reduction in FAR using the 2nd stage classifier for the 4×896 monophone background based wake word system. Apart from the accuracy gain, computational resources required by the 2nd stage classifier is very cheap compared to the 1st stage. Its size is more than 100 times smaller than the 1st stage model and there is no need to run Viterbi decoding on a restricted graph.

5. CONCLUSION

In this paper, we introduce a two-stage wake word system. We propose a different way to model the non-keyword background audio by expanding the speech/non-speech events to more specific monophone-based units at the first stage and we also present how to extract richer features for a second stage verification. By using the new wake word system, we can reduce FRR by about 16% relatively when the false alarm level is maintained, and we are able to reduce FAR by about 37% relatively if we maintain the level of miss rate. We also demonstrate effectiveness of the second stage classifier itself. It is able to reduce the FAR by about 67% relatively on top of 1st stage hypothesis with very few computational resources. In the future, it is worthwhile to apply sequence-discriminative training [26] [27] which is widely used in the field of LVCSR to train the wake word acoustic model, since we could now have rich phoneme-based information in the lattice.

### Table 1: Summary of different wake word systems

<table>
<thead>
<tr>
<th></th>
<th>2nd NN: 2×64</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FRR (fix FAR=2y)</td>
<td>FAR (fix FRR=0.04)</td>
<td>Params</td>
</tr>
<tr>
<td>SP/NSP(4×896)</td>
<td>0.051</td>
<td>3.71y</td>
<td>3.02M</td>
</tr>
<tr>
<td>SP/NSP(4×1024)</td>
<td>0.050</td>
<td>3.43y</td>
<td>3.84M</td>
</tr>
<tr>
<td>monophone(4×896)</td>
<td>0.043</td>
<td>2.35y</td>
<td>3.15M</td>
</tr>
<tr>
<td>monophone(4×1024)</td>
<td>0.042</td>
<td>2.31y</td>
<td>3.99M</td>
</tr>
</tbody>
</table>

Figure 3: End-to-end comparison of the baseline wake word system and the new system using monophone-based background modeling; DET curves on test set; Axis of the false alarm rate is obscured due to the sensitive nature of this information.
6. REFERENCES


[10] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Björn Hofmeister, and Shiv Vitaladevuni, “Multi-task learning and weighted cross-entropy for dnn-based key-


[14] Jinx Guo, Kenichi Kumatani, Ming Sun, Minhua Wu, Anirudh Raju, Nikko Strom, and Arindam Mandal, “Time-delayed bottleneck high-


[27] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Gahramani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, “Purely sequence-trained neural networks for asr based on lattice-

free mmi.,” Submitted to Interspeech, 2016.