CONTEXT-DEPENDENT POINT PROCESS MODELS FOR KEYWORD SEARCH AND DETECTION-BASED ASR

Chunxi Liu[⋆] *Aren Jansen*[⋆†] *Sanjeev Khudanpur*[⋆†]

[⋆] Center for Language and Speech Processing & Department of Electrical and Computer Engineering
[†] Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD USA
{chunxi, aren, khudanpur}@jhu.edu

ABSTRACT

The point process model (PPM) for keyword search (KWS) is a whole-word parametric approach that characterizes each query type by the timing of phonetic events observed during its production. In this paper, we first extend the PPM modeling framework to operate on context-dependent phonetic event patterns instead of monophone patterns considered in the past, which provides significant KWS improvements. Second, we use the context-dependent PPMs to drive a detection-based speech recognition architecture thats runs parallel word detectors covering the whole vocabulary and uses the independent detections to construct lattices that can be used for both KWS indexing and LVCSR decoding. This strategy produces significant improvements over the original PPM KWS framework and provides an encouraging first attempt at PPM-based LVCSR.

***Index Terms*—** point process model, keyword search, speech recognition

1. INTRODUCTION

Originally proposed in [1], the point process model (PPM) for keyword search (KWS) is a parametric approach that assumes observed phonetic events derived from the input speech signal are generated by underlying keyword-specific Poisson processes [1]. A series of past efforts have been focused to improve the model estimation and search algorithms [2, 3, 4], leading to recent demonstration of state-of-the-art standalone phonetic search performance in both high- and low-resource settings with substantial complementarity with high-performance LVCSR-based systems [5, 6, 7].

However, the past comprehensive benchmark evaluations have been limited to building the PPM search index and parametric models on monophone event patterns without considering the cross-phone context, in contrast to common practices employed by context-dependent (triphone) HMM-based automatic speech recognition (ASR) systems [8]. [9] is the only related work of using acoustic event patterns beyond monophone detectors, where untied states of whole-word HMM-GMM acoustic models were used to define the detector set. However, that work considered only a small vocabulary digit recognition task that required many examples of each word in the lexicon. In this paper, we exploit state-of-the-art deep neural network (DNN) acoustic models to generate the tied triphone state (senone) events, which enable the application of dictionary-based PPMs and subsequent maximum a posteriori (MAP) estimation for scaling to open vocabulary search tasks.

This work was partially supported by NSF Grants No IIS 1005411, No IIS 0963898 and DARPA Contract No HR0011-15-2-0024.

In addition to open vocabulary search, we also consider the use of our context-dependent PPMs for large vocabulary continuous speech recognition (LVCSR), which is possible due to recent advances in the computational efficiency of PPM search algorithms. We employ the detection-based ASR framework previously considered for small vocabulary tasks [10, 9]. In contrast to the Viterbi search of HMM systems, this alternative approach applies a set of parallel word detectors and derives the most likely word sequence from their combined output. Critical to this process is the construction of a word lattice from the set of independent word detections so that language models can be subsequently applied. We first adapt the confusion network (CN) [11] algorithm as our baseline approach and propose our own lattice construction algorithm specially designed for the PPM framework. Both data structures can be then composed with a finite state transducer (FST) based language model and either decoded for LVCSR or used as the keyword search index for in-vocabulary queries. We evaluate our proposed approaches with comprehensive KWS and LVCSR experiments under the IARPA Babel Program framework [7], which aims to develop robust low-resource techniques to facilitate KWS search on massive multilingual speech corpus. We find incorporating context-dependency into the PPM framework produces substantial improvements over the original monophone PPM system and demonstrate reasonable LVCSR performance with a small computational footprint.

2. POINT PROCESS MODELS FOR KWS

The PPM KWS framework first transforms input speech signals into smoothed phone posteriorgram trajectories, of which each local maxima above a threshold is identified as a phonetic event corresponding to a single phone occurrence [2]. The extracted phonetic events form a phonetic index. Formally, given a time interval $(t, t+T]$, for each phone p in phone set $\mathcal{P}$, we denote its phonetic event set in time at which phone p occurs relative to time t as $N_p = \{t_1, \ldots, t_{n_p}\}$, where n_p is the total number of events within $(t, t+T]$. Thus, the set of all observed events arriving in $(t, t+T]$ is $O_{t,t+T} = \{N_p\}_{p\in\mathcal{P}}$.

Given a keyword w with its occurrence time t and duration T, the arrival of phonetic events during the given word realization is modeled as a collection of inhomogeneous Poisson processes, one per phone. We approximate the continuous Poisson rate function in interval $(t, t+T]$ as a piecewise constant function over D uniformly spaced divisions, with the inhomogeneous rate parameter for phone p denoted as $\lambda_{p,d}$ for $d = 1, ..., D$. We denote the set of keyword-specific model parameters as θ_w, and the likelihood of the entire collection $O_{t,t+T}$ under θ_w given T can be expressed as

$$p(O_{t,t+T}|T,\theta_w) = \prod_{p\in P}\prod_{d=1}^{D}(\lambda_{p,d})^{n_{p,d}}e^{-\lambda_{p,d}T/D} \quad (1)$$

PPM makes the assumption that the phonetic event timing distribution is independent of actual word duration, and thus linearly normalizes all arrival times within $(t, t+T]$ into the interval $(0, 1]$ to produce a transformed event set $O'_{t,t+T}$. Thus, after a change of variables, the likelihood function of Eq. 1 with $O'_{t,t+T}$ becomes

$$p(O'_{t,t+T}|T,\theta_w) = \prod_{p\in P}\prod_{d=1}^{D}(\lambda_{p,d})^{n_{p,d}}e^{-\lambda_{p,d}/D} \quad (2)$$

The phonetic event distribution (i.e., the time-varying Poisson rate function) of each phone instance within a word can be modeled by a single Gaussian distribution, and given the dictionary, a PPM can be constructed by assigning a Gaussian to each phone in the pronunciation [4]. Each Gaussian is further transformed to a GMM to account for phone confusions, where the mixture weights can be estimated over entire corpus. For example, a dictionary model for the Haitian word "alo" is shown in Figure 1(1). Further, the GMMs are updated by maximum a posteriori (MAP) estimation, benefiting from the observed phonetic event timing information of any available training examples [4]. The resulting MAP updated model for "alo" is depicted in Figure 1(2).

The PPM also requires a background model, assuming that outside the word of interest, phonetic events are produced by a homogeneous Poisson process characterized by one independent rate parameter μ_p for each phone p. Thus, the likelihood of observation $O_{t,t+T}$ under the background model with parameters θ_{bg} is obtained as

$$p(O_{t,t+T}|T,\theta_{bg}) = \prod_{p\in P}(\mu_p)^{n_p}e^{-\mu_p T} \quad (3)$$

To evaluate an unknown utterance, we define the keyword detection function $d_w(t)$ as the log-likelihood ratio of phonetic events under the keyword and background model given by

$$\begin{aligned} d_w(t) &= \log\left[\frac{P(O_{t,\infty}|\theta_w)}{P(O_{t,\infty}|\theta_{bg})}\right] \\ &= \log\left[\int_0^\infty \frac{p(O'_{t,t+T}|T,\theta_w)P(T|\theta_w)}{T^{|O_{t,t+T}|}p(O_{t,t+T}|T,\theta_{bg})}dT\right] \\ &\approx \max_T \log\left[\frac{p(O'_{t,t+T}|T,\theta_w)P(T|\theta_w)}{T^{|O_{t,t+T}|}p(O_{t,t+T}|T,\theta_{bg})}\right] \end{aligned} \quad (4)$$

where the hypothesis keyword duration T is a latent variable modeled by a gamma distribution, and the integral can be approximated by computing over a number of candidate durations and taking the max (with the corresponding T as the hypothesized duration) [3].

3. CONTEXT-DEPENDENT PPMS

3.1. Deriving context-dependent phonetic events from DNN

To generate the context-dependent phonetic event streams, we use a state-of-the-art DNN acoustic model generated with the Kaldi toolkit [12]. We take as our events the set of tied triphone HMM states (senones), which are derived from traditional decision tree clustering of triphone states [8]. The DNN forward pass produces posteriorgrams over the senones which provide the input to the PPM pipeline described above, but where the monophone category set $\mathcal{P}$ is now replaced with the set of senones. The PPM search index is created by filtering the posteriorgrams according to the empirical distribution of each senone's duration and extracting the local maxima exceeding an empirically assigned threshold [2].

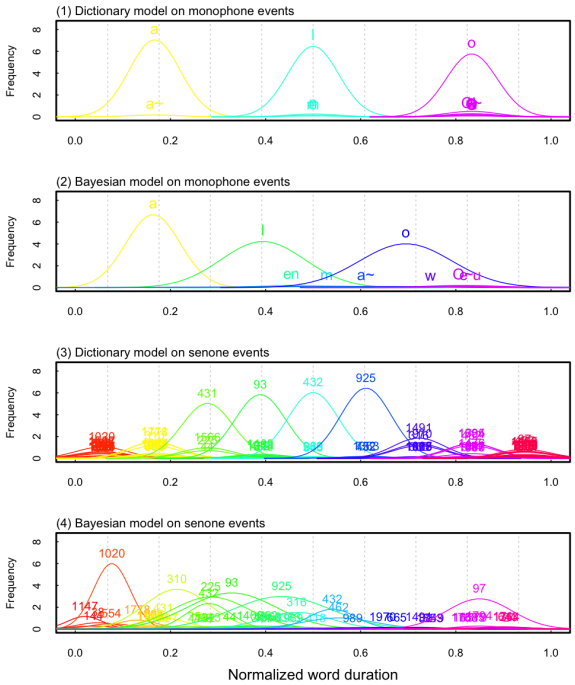

Fig. 1. Dictionary/Bayesian MAP estimated phone timing models for the keyword "alo", based on monphone/senone events.

3.2. Context-dependent extensions to PPM construction

The original dictionary PPM is constructed by the monophone sequence provided by the pronunciation lexicon, so now we need to extend the dictionary form to that based on triphones, and construct the dictionary PPM based on the senone sequence. Given the left and right context phones, we can obtain the senone index for each central phone by answering the questions in phonetic decision tree. However, for the first and last phones of a single keyword the left and right context phones, respectively, are unknown without identifying the adjacent words. Thus, we assume that each phone in the phone set is equally likely to be the unknown context phones and we accumulate the senone index count by considering all these possibilities. We normalize each senone index count to determine the senone probability that is subsequently used as the GMM mixture weight in that position. Finally, we smear and renormalize the mixture weights using a global senone confusion matrix estimated from the training corpus. The resulting dictionary PPM of word "alo" consisting of senones indexed by integers is shown in Figure 1(3). Maximum a posteriori (MAP) estimation including any training instances of the word is subsequently performed using the observed senone event streams. The MAP-estimated PPM for "alo" is shown in Figure 1(4), where we see substantial movement of the senone timing distributions.

4. PPM-BASED LATTICE CONSTRUCTION FOR KWS AND DETECTION-BASED LVCSR

Our proposed detection-based ASR architecture consists of four steps: (i) we build a PPM for each in-vocabulary (IV) unigram word, (ii) for each test utterance, run parallel word detectors for the whole vocabulary, (iii) use the resulting independent word detections to build confusion networks (CNs) or word lattices, and (iv) use standard techniques to process the CNs/lattices for KWS indexing [13] and LVCSR decoding [14]. Below we describe our

CN and lattice construction methodologies.

4.1. PPM-based confusion network construction

The standard confusion network (CN) is derived from a decoding lattice as a more compact representation with relaxed word sequence constraints [11]. It requires that the posterior probability for each arc in the lattice is estimated (by running forward-backward algorithm), and that the temporal partial order between arcs is derived based on lattice topology. Since there are word identity, start time, duration, and posterior probability estimates (by a logistic regression applied to the likelihood ratio detection score of Eq. 4) associated with each PPM detection, we can naturally adapt the algorithm of [11] to build CNs based on PPM detections rather than decoding lattices. For each test utterance, we first sort the PPM detections of all the IV words according to their start time, and initialize each detection as an equivalence class (formed by word identity, start and end times). Second, we perform intra-word clustering to merge the equivalence classes of the same word identity, and then perform inter-word clustering based on phonetic similarity, resulting in a complete alignment of competing detections as confusion bins.

4.2. PPM-based lattice generation

The duration of a PPM detection is hypothesized as the one that gives the maximum detection function value of Eq. 4, which may not be as accurate as that derived from the HMMs based on frame likelihood. Since the KWS scoring metrics can accommodate small time differences between the detections and the true references, such approximated duration from PPM is generally sufficient for the KWS task. However, the CN algorithm relies on strict temporal order between word components for clustering and inaccurate durations can lead to suboptimal results. Moreover, the CN algorithm requires word posterior estimates for each detection; the raw PPM detection score is a likelihood ratio and applying a global logistic regression for normalization is known to give suboptimal posterior estimates [6]. Therefore, we propose a lattice construction algorithm for the PPM framework to accommodate the duration uncertainties and rely on word acoustic likelihood only, as described below.

First, for each PPM detection, we express its joint likelihood of acoustic observations $O_{t,t+T}$ and hypothesized duration as

$$p(O_{t,t+T}, T|\theta_w) = p(O_{t,t+T}|T, \theta_w)P(T|\theta_w) \tag{5}$$

where $p(O_{t,t+T}|T, \theta_w)$ is given by Eq. 1 and further by Eq. 2 with the event set normalized in time, and $P(T|\theta_w)$ is a word-specific gamma distribution. Second, for an arbitrary region between two word detections, e.g. non-speech silence or noise, we employ a separate silence model of homogeneous Poisson process for the observed acoustic events in that region that takes the form

$$\begin{aligned} p(O_{t,t+T}, T|\theta_{sil}) &= p(O_{t,t+T}|T, \theta_{sil})P(T|\theta_{sil}) \\ &= \prod_{p\in P} (\mu_p)^{n_p} e^{-\mu_p T} P(T|\theta_{sil}) \end{aligned} \tag{6}$$

where p represents either context-independent monophone or context-dependent senone in the event set $\mathcal{P}$, μ_p is the homogeneous Poisson rate parameter for each p under the silence model θ_{sil} with $P(T|\theta_{sil})$ modeled by a gamma distribution. Thus, we have approaches to compute acoustic likelihoods given any word hypothesis or an arbitrary region of acoustic observations.

Our strategy is to define "words-on-nodes" lattices, where each word detection becomes a node and the edges encode the temporal sequence of detections with directed arcs that can accommodate a sensible amount of temporal overlap. We define the construction process using the following notation. We denote the set of all the detections within a given utterance as $\mathcal{D}$, and sort $\mathcal{D}$ according to each detection's start time. For each word detection $d_i \in \mathcal{D}$ with index i in time, we define a node with acoustic likelihood given by Eq. 5, and $t_s(d_i)$ as its start time. We refer to all observed acoustic events that have arrived during the course of d_i as set $\rho(d_i)$, which is also the set of events used to give the maximum value of Eq. 4.

The goal is to produce a directed acyclic graph, where $\phi(d_i)$ is the set of word detections (nodes) that d_i has an outgoing edge to, such that any word in $\phi(d_i)$ can follow d_i in the output word sequence. We make each detection d_i (except the final node defined as the end of the utterance) connect to at least one another next node (in time) d_j ($j > i$), which we require by that: (i) d_j does not consume any acoustic event arrived during d_i, i.e., no intersection between $\rho(d_j)$ and $\rho(d_i)$, and (ii) the time gap between observations of d_i and d_j does not exceed a maximum allowable time gap δ (initialized as 1 sec) if possible. If we denote $t'_s(d_i)$ as time of the first phonetic event observed in time within d_i, and $t'_e(d_i)$ as time of its last observed event, then condition (i) becomes $t'_e(d_i) < t'_s(d_j)$, and condition (ii) becomes $(t_s(d_j) - t'_e(d_i)) < \delta$.

Also, if there are no acoustic events between time interval $(t'_e(d_i), t_s(d_j))$, we connect d_i to d_j with a free edge. If there is, we add a new node as d_{sil} of which the acoustic likelihood is computed by Eq. 6 on the acoustic events between interval $(t'_e(d_i), t_s(d_j))$ and the duration is given by $(t_s(d_j) - t'_e(d_i))$; further, we connect d_i to d_{sil} and connect d_{sil} to d_j.

In this approach, we can finally obtain a directed acyclic graph where each node is associated with its word identity, acoustic likelihood, start time and duration. By replying on the phonetic event timing information to determine the temporal order of the word sequence, we relax the accurate estimation of word start and end times but still enable an appropriate lattice construction, with the unidentified phonetic events accounted by optionally added silence nodes.

Finally, we convert the graph into a standard lattice with word and acoustic likelihood on each arc, which can be processed by standard FST-based algorithms such as language model composition.

5. EXPERIMENTS

5.1. Evaluation design and system implementation

We perform evaluation in the same IARPA Babel Program framework as described in [6], on two of the Babel languages – Haitian[1] and Bengali[2] under the limited language (LimitedLP) resource condition. For training, each language contains 10 hours of transcribed speech audio along with language model text and pronunciation dictionary entries restricted to those in the given 10 hours. For evaluation, we have a 10-hour development-testing collection for each language while tuning on a 2-hour subset. We use Actual Term-Weighted Value (ATWV) and Oracular Term-Weighted Value (OTWV) [15] as KWS scoring metrics. The same infrastructure of DNN-HMM acoustic models is used as [6], with a 5-layer DNN of p-norm units ($p = 2$) [16] and pitch-augmented PLP features. Each dictionary-based unigram or multi-word PPM is synthesized and MAP updated as in [4, 5], with the extension described in Section 3 when operating on senone events.

[1]Language collection release `IARPA-babel201b-v0.2b`.

[2]Language collection release `IARPA-babel103b-v0.4b`.

Table 1. *PPM search performance for Haitian and Bengali, along with relative gain from using senone over monophone events.*

Language	PPM System	OTWV (All)	ATWV (All)	ATWV (IV unigram)	ATWV (unigram)	ATWV (multiword)
Haitian	# of keywords	1921	1921	418	573	1348
	monophone	0.361	0.212	0.119	0.127	0.249
	senone	0.380	0.225	0.158	0.159	0.253
	% Gain	5.3	6.1	32.8	25.2	1.6
Bengali	# of keywords	1967	1967	603	926	1041
	monophone	0.222	0.101	0.029	0.041	0.154
	senone	0.237	0.111	0.061	0.061	0.155
	% Gain	6.8	9.9	110.3	48.8	0.6

Table 2. *KWS performance (IV unigrams) comparison between keyword-specific PPM search and lattice-based approach.*

Language	PPM System	OTWV	ATWV
Haitian	baseline, monophone	0.241	0.119
	cn, monophone	0.233	0.066
	lattice, monophone	0.257	0.129
	% Gain	6.6	8.4
	baseline, senone	0.298	0.158
	lattice, senone	0.305	0.175
	% Gain	2.3	10.8
Bengali	baseline, monophone	0.113	0.029
	lattice, monophone	0.122	0.029
	% Gain	8.0	0.0
	baseline, senone	0.162	0.061
	lattice, senone	0.173	0.080
	% Gain	6.8	31.1

Table 3. *WER performance from PPM and HMM lattices.*

Language	System	WER
Haitian	PPM lattice, monophone	74.1
	PPM lattice, senone	69.8
	HMM, senone	59.6
Bengali	PPM lattice, monophone	80.5
	PPM lattice, senone	77.9
	HMM, senone	66.8

5.2. Evaluation for Context-dependent PPM

We first evaluate the efficacy of incorporating context-dependency into the original PPM framework without lattice construction, where the word posterior is approximated by a logistic regression applied to detection score of Eq. 4. The results are shown in Table 1. We see that context-dependent PPM on senone events significantly outperforms the monophone baseline in nearly all categories, but remains the same for multi-word keywords. We can account for this by the fact that more monophone events are observed in the generally longer multiword queries, which limits the additional benefit of more detailed triphone patterns.

Finally, it is important to note that even though the senone set (approximately 2000 units) is much larger than monophone set ($\sim$50 dimension), in practice the PPM search index size is on average only 2.2 times larger than before. This is a result of the fact that the increase in posteriorgram units does not substantially reduce event sparsity since the new units are generally mutually exclusive. It follows that the PPM's storage advantages highlighted in [5] are maintained despite the increased model detail.

5.3. Evaluation for PPM-based lattice generation

We refer to the independent keyword-specific PPM search evaluated above (without lattice construction) as the baseline in Table 2, and compare with the PPM's CN and lattice-based KWS. Since keywords tend to have lower unigram probabilities in training transcript, to increase the keyword recall we keep more detections for words that occur rarely during training. To accomplish this we prune PPM detections of each IV unigram based on its unigram probability using empirically determined thresholds. Further, confusion networks and lattices are obtained as described in Section 4, and we compose them with a FST-based language model to give each arc a trigram language model prior, with a tuned acoustic scaling factor.

Table 2 shows that the adapted confusion network approach does not outperform baselines, a result of suboptimal duration and posterior estimation issues discussed in Section 4.2. The proposed words-on-nodes lattice generation algorithm leads to consistent KWS improvements for both monophone and senone event-based systems. We also find that, combining context-dependency and PPM lattice generation yields significant gains over the original monophone baseline.

Finally, Table 3 shows the lattices generated by PPM framework can also provide reasonable ASR performance. Though its WER trails the DNN-HMM systems, it has obvious computational merit. The PPM index is created about 2x faster than real time (RT), and each IV word can be detected in parallel with speeds 500,000x faster than RT [3]. The subsequent PPM lattice construction complexity is of order $O(N^2)$, where N is the number of detections in an utterance; since we only consider connecting each detection with its close neighbors, the runtime in practice is in excess of 1,000x faster than RT. Thus, we find the overall runtime of PPM decoding and lattice generation much more efficient than the real-time factor 8.41 of the DNN-HMM based lattice generation (comparing based on one single core of a 2.40-GHz Intel Xeon processor). The subsequent operations of language model composition and lattice indexing are efficiently implemented in a WFST-based framework as before [13].

6. CONCLUSIONS

The incorporation of context-dependent phonetic events into the PPM framework produces substantial KWS performance improvements with only a small increase in computational complexity. Lattice generation produces further KWS improvements by incorporating language models and better score normalization. Furthermore, lattices support LVCSR decoding, which gives reasonable performance for a first attempt on a difficult task.

7. REFERENCES

[1] Aren Jansen and Partha Niyogi, "Point process models for spotting keywords in continuous speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1457–1470, 2009.

[2] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "Event selection from phone posteriorgrams using matched filters.," in *Proc. INTERSPEECH*, 2011.

[3] Keith Kintzley, Aren Jansen, Kenneth Church, and Hynek Hermansky, "Inverting the point process model for fast phonetic keyword search," in *Proc. INTERSPEECH*, 2012.

[4] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "MAP estimation of whole-word acoustic models with dictionary priors," in *Proc. INTERSPEECH*, 2012.

[5] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "Featherweight phonetic keyword search for conversational speech," in *Proc. ICASSP*, 2014.

[6] Chunxi Liu, Aren Jansen, Guoguo Chen, Keith Kintzley, Jan Trmal, and Sanjeev Khudanpur, "Low-resource open vocabulary keyword search using point process models," in *Proc. INTERSPEECH*, 2014.

[7] Jan Trmal, Guoguo Chen, Dan Povey, Sanjeev Khudanpur, Pegah Ghahremani, Xiaohui Zhang, Vimal Manohar, Chunxi Liu, Aren Jansen, and Dietrich Klakow, "A keyword search system using open source software," in *IEEE SLT*, 2014.

[8] Steve J Young, Julian J Odell, and Philip C Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.

[9] Aren Jansen and Partha Niyogi, "Detection-based speech recognition with sparse point process models," in *Proc. ICASSP*, 2010.

[10] Petr Fousek and Hynek Hermansky, "Towards ASR based on hierarchical posterior-based keyword recognition," in *Proc. ICASSP*, 2006.

[11] Lidia Mangu, Eric Brill, and Andreas Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[12] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[13] Guoguo Chen, Sanjeev Khudanpur, Daniel Povey, Jan Trmal, David Yarowsky, and Oguz Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in low resource languages," in *Proc. ICASSP*, 2013.

[14] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[15] NIST, "The Spoken Term Detection (STD) 2006 Evaluation Plan," *http://www.nist.gov/speech/tests/std/*, 2009.

[16] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*, 2014.