On the Use of Acoustic Unit Discovery for Language Recognition

Stephen H. Shum, *Student Member, IEEE*, David F. Harwath, *Student Member, IEEE*, Najim Dehak, *Senior Member, IEEE*, and James R. Glass, *Fellow, IEEE*

***Abstract*—In this paper, we explore the use of large-scale acoustic unit discovery for language recognition. The deep neural network-based approaches that have achieved recent success in this task require transcribed speech and pronunciation dictionaries, which may be limited in availability and expensive to obtain. We aim to replace the need for such supervision via the unsupervised discovery of acoustic units. In this work, we present a parallelized version of a Bayesian nonparametric model from previous work and use it to learn acoustic units from a few hundred hours of multilingual data. These unit (or senone) sequences are then used as targets to train a deep neural network-based i-vector language recognition system. We find that a score-level fusion of our unsupervised system with an acoustic baseline can shrink the gap significantly between the baseline and a supervised benchmark system built using transcribed English. Subsequent experiments also show that an improved acoustic representation of the data can yield substantial performance gains and that language specificity is important for discovering meaningful acoustic units. We validate the generalizability of our proposed approach by presenting state-of-the-art results that exhibit similar trends on the NIST Language Recognition Evaluations from 2011 and 2015.**

***Index Terms*—Acoustic unit discovery (AUD), bottleneck features, deep neural networks (DNNs), i-vector, language recognition, senone posteriors.**

I. Introduction

The effectiveness of deep neural networks (DNNs) for automatic speech recognition (ASR) [1] has led to their use in other speech-related classification tasks, including speaker and language recognition [2]–[9]. One of the reasons for their success can be attributed to the "phonetic awareness" of the trained DNNs and their corresponding feature space [4]. The training of such DNNs, however, relies on the presence of pronunciation dictionaries and large amounts of transcribed speech, which may only be available for a small subset of the languages present in the evaluation task. For example, the work in [6], [7] used only transcribed English from the Switchboard I corpus [10] to build a system that could distinguish between 24 different languages, while the use of transcribed data from additional languages achieved even better results in [8]. On the other end of the spectrum, this brings to bear the question of how well we can do without any transcribed data. Following the premise of [11], we aim to exploit the existing sound pattern structure of speech without the need for transcription or a dictionary. In this paper specifically, we investigate the effect of unsupervised acoustic unit discovery (AUD) on language recognition.

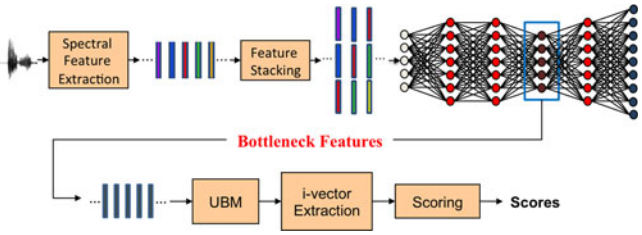


Fig. 1. An overview of a BN i-vector system: stacked spectral features are passed as input to a neural network, whose activations at a BN layer are used as features for an i-vector classification system. The resulting i-vectors are a low-dimensional summary of an utterance's distribution of BN features. (Adapted from Fig. 1 of [6].)

To do so, we follow the framework proposed in [2], [5]–[7], where a DNN is trained from spectral input features and ASR-based output labels such that the activations at a so-called bottleneck (BN) layer provide frame-level features of manageable dimensionality. A BN feature of a given frame of audio can be seen as a compression of the information about both the frame's phonetic class and context [5]. These features can then be treated as acoustic features of their own, from which an i-vector system can be built for language recognition [12]. Fig. 1 presents an overview of this system.

In our experimental setup, we follow most closely the work in [6], [7], which proposed a unified DNN framework for both speaker and language recognition. While this work focuses on language recognition, we adopt the same DNN architecture and set of labeled data as [6], [7] for the sake of comparison and consistency. The work in [5] proposes the use of stacked BN features for language recognition where two DNNs are cascaded: the input of the second DNN consists of stacked BN features from the first DNN. And as an alternative to BN features, other approaches involving DNNs and their output (i.e., senone) posteriors have been explored for language recognition in [3], [9] as well as for speaker recognition [4].

In this paper, unlike any recent work on language recognition to the best of our knowledge, we replace the ASR-based output labels from the original DNN-based setup with those learned from a Bayesian nonparametric model that learns an appropriate set of sub-word units automatically from speech data [13].

Manuscript received March 15, 2016; revised May 28, 2016; accepted June 06, 2016. Date of publication June 20, 2016; date of current version July 14, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong Cui.

S. H. Shum, D. F. Harwath, and J. R. Glass are with the Computer Science and Artificial Intelligence Laboratory Massachusetts Intitute of Technology, Cambridge, MA 02139 USA (e-mail: sshum@csail.mit.edu; dharwath@csail.mit.edu; glass@csail.mit.edu).

N. Dehak is with the Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: ndehak3@jhu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2016.2582260

The development of this model was motivated by the desire for robust zero resource speech technologies that can operate without the expert-provided linguistic knowledge that standard ASR systems rely on [14]. Designed to uncover phone-like units from a given language, the resulting AUD system simultaneously segments the speech, discovers a proper set of sub-word units, and learns a Hidden Markov Model (HMM) for each [13]. Using neither transcribed data nor prior language-specific knowledge, this system obtained results on TIMIT that demonstrate the ability to discover sub-word units that are highly correlated with English phones, produced a better segmentation than the state-of-the-art unsupervised baseline, and performed well on a spoken term detection task [13].

Despite the promise of this model and that of similar systems [11], [15], [16], we are still unable to robustly and precisely uncover a particular language's phonetic inventory. In this paper, we chose to broaden our consideration of unsupervised unit discovery from a monolingual setting to a multilingual one. Instead of focusing on any single language in particular, we aim to learn a set of acoustic units from many different languages at once. To paraphrase the analogy to human infants, who must specialize their speech perception and production systems to their native language (though perhaps with help from other sensory modalities) [14], we see our human infant as developing in a multilingual household. And more importantly, because we are not bound by any limited quantity of transcribed corpora, our models can instead be built on as much data as they can handle. Indeed, unsupervised methods give us the flexibility to work directly on data that matches the test domain,[1] thus avoiding issues of language or channel mismatch.

In addition to the work reviewed in [14], the notion of transforming speech and audio data into a sequence of arbitrary symbols has been well-explored [17]. The work in [11] details the unsupervised training of an HMM-based self-organizing unit recognizer, while the work in [18] learns a set of "acoustic unit descriptors" to represent audio content for event classification and detection. The work in [19]–[21] proposes the Automatic Language Independent Speech Processing approach, which was initially developed for low bit-rate speech coding before evolving into a generic method for audio indexing, retrieval, and recognition, including initial attempts at speaker verification and forgery, as well as language identification [20].

The unsupervised tokenization of speech for language identification is also a problem that has been explored in the past. Whereas the previously mentioned approaches using i-vectors constitute an acoustic approach to language identification, success has also been achieved using phonotactic approaches [22], which typically involve a high quality phoneme recognizer for speech tokenization [23]. A full review of approaches in phonotactic language recognition is beyond the scope of this paper, but the initial work in [24] developed a system using a Gaussian Mixture Model (GMM) for per-frame tokenization that, without requiring prior transcribed speech material, performed competitively against state-of-the-art tokenizers at a lower computational cost. Subsequently, the work in [25] introduced an bootstapped learning procedure to learn a set of HMM-based acoustic segment models (ASMs) from an initial multilingual phone inventory and adopts a phonotactic approach to language identification using a vector space model of acoustic unit co-occurrence statistics. While these ASMs are analogous to the acoustic units that we propose to discover in this work, our approach to language recognition further deviates from that of [25] in the use of BN features for an acoustic i-vector system.

Although more detailed explanations can be found throughout the rest of this paper, let us first summarize the novel contributions and findings of our work below:

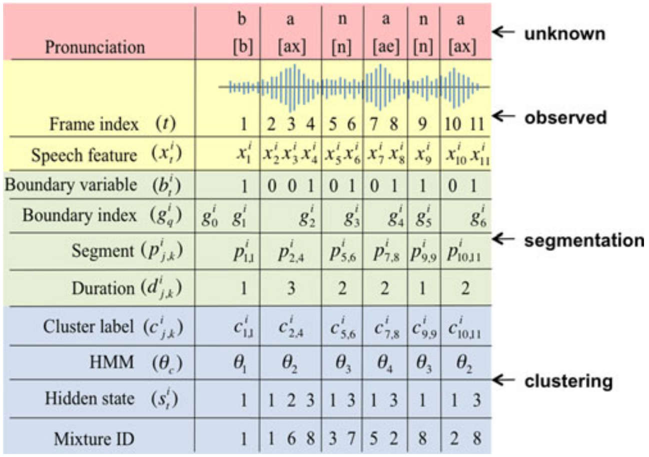
1) We show that a system built from learned acoustic units can be used effectively for language recognition. In particular, we find in Section IV-F that a score-level fusion with a baseline system built from acoustic features yields substantial gains and significantly closes the gap between the acoustic feature baseline and a benchmark system built using transcribed English, suggesting that AUD provides complementary information to that of a bag-of-features baseline.
2) We then find in Section IV-G that using an improved representation of speech (i.e., supervised BN features) as input to our AUD system can yield acoustic units that similarly improve performance on our language recognition task, and additional score-level fusion provides even further gains. This continues to motivate the need for a better understanding of the speech signal.
3) We demonstrate the ability to learn acoustic units in an unsupervised fashion on a dataset containing hundreds of hours of speech. As described in Section II, this was achieved by modifying a Bayesian nonparametric model in a way that allows for effective parallelization. To the best of our knowledge, this is also the first Kaldi-based implementation of the AUD process [26].
4) We present our initial results on the Language Recognition Evaluation (LRE) from 2011 [27] presented by the National Institute of Standards and Technology (NIST) and subsequently validate the generalizability of our proposed approach in Sections V-B and V-C on the NIST 2015 LRE, which features a modified evaluation protocol involving specific language clusters.

The rest of this paper is organized as follows. Section II outlines our unit discovery process, and we reiterate the language recognition system from [6] that serves as our experimental framework in Section III. Section IV presents our initial results; Section V discusses some of the design choices that both worked and didn't work, and validates our original results on another dataset. Finally, we conclude in Section VI with a look ahead to future work.

II. Acoustic Unit Discovery

In this section, we outline the essentials of a previously-proposed AUD process [13] and highlight the modifications we

[1] To be sure, this does not mean that we are training our models on the test data; rather, because we are not bound to the data for which we have transcription labels, we can work more directly on the provided training data that pertains more closely to the evaluation task at hand.



Pronunciation	b [b]	a [ax]	n [n]	a [ae]	n [n]	a [ax]
Frame index (t)	1	2 3 4	5 6	7 8	9	10 11
Speech feature (x_t^i)	x_1^i	$x_2^i x_3^i x_4^i$	$x_5^i x_6^i$	$x_7^i x_8^i$	x_9^i	$x_{10}^i x_{11}^i$
Boundary variable (b_t^i)	1	0 0 1	0 1	0 1	1	0 1
Boundary index (g_q^i)	g_0^i g_1^i	g_2^i	g_3^i	g_4^i	g_5^i	g_6^i
Segment $(p_{j,k}^i)$	$p_{1,1}^i$	$p_{2,4}^i$	$p_{5,6}^i$	$p_{7,8}^i$	$p_{9,9}^i$	$p_{10,11}^i$
Duration $(d_{j,k}^i)$	1	3	2	2	1	2
Cluster label $(c_{j,k}^i)$	$c_{1,1}^i$	$c_{2,4}^i$	$c_{5,6}^i$	$c_{7,8}^i$	$c_{9,9}^i$	$c_{10,11}^i$
HMM (θ_c)	θ_1	θ_2	θ_3	θ_4	θ_3	θ_2
Hidden state (s_t^i)	1	1 2 3	1 3	1 3	1	1 3
Mixture ID	1	1 6 8	3 7	5 2	8	2 8

Fig. 2. An example of the observed data and hidden variables in the AUD model, modified directly from Fig. 1 of [13].

make in its updated implementation to make inference more computationally feasible on data involving hundreds of hours of speech.

A. A Bayesian Nonparametric Model

Given a set of spoken utterances, the goal of AUD is to jointly learn the following [28]:

1) segmentation—find the phonetic boundaries within each utterance;
2) clustering—obtain an appropriate number of clusters within which acoustically similar segments can be grouped;
3) modeling—learn a HMM to model each sub-word acoustic unit.

In [13], all three sub-tasks were modeled using latent variables in a single Bayesian nonparametric model. More specifically, [13] formulates a Dirichlet process mixture model where each mixture is a HMM used to model a sub-word unit and generate observed segments of that unit. Via Gibbs sampling inference, the model seeks to obtain the set of sub-word units, segmentation, clustering, and HMMs that best represent the observed data.

An explanation of the associated variables and the entire generative process, as well as a derivation of the conditional posterior distributions for each hidden variable in the model is provided in [13]. Fig. 2 directly replicates an example of the observed data and hidden variables of the setup [13]; we outline the essential ingredients below:

1) speech feature (x_t)—13-dimensional MFCCs and their first- and second-order derivatives extracted every 10 ms, resulting in a 39-dimensional observed feature vector;
2) boundary (b_t)—a binary variable indicating whether a phone boundary exists $(b_t = 1)$ between x_t and x_{t+1} or not $(b_t = 0)$;
3) HMM (Θ_c)—each HMM has three emission states, corresponding respectively to the beginning, middle, and end of each sub-word unit. A traversal of each HMM must start from the first (left-most) state, and transitions may only occur from left to right. While skipping of the middle and last states is allowed in [13], our implementation requires that each segment be at least three frames in length. The emission probability of each state is modeled by a GMM.
4) hidden state (s_t)—the hidden state index of the HMM associated with each feature vector, x_t.
5) mixture ID (m_t)—the Gaussian mixture index associated with each feature vector, x_t.

If we assume, for the time being, that the values of the boundary variables, b_t, are given, then the generative process looks as follows:

1) Given a segment, $p = \{x_t | L < t \leq R\}$, as determined by two boundary variables $(b_L, b_R = 1)$, choose a cluster label, $c \in \mathcal{C}$, which can either be an existing label or a new one. (The Dirichlet process allows for a potentially infinite number of clusters.) This cluster label will determine which HMM, Θ_c, is used to generate the segment.
2) Given the HMM corresponding to the cluster label, choose a hidden state, s_t, for each feature vector in the segment.
3) Given the hidden state of each feature vector, choose a mixture from the GMM of the chosen state, m_t.
4) Given the mixture ID, generate the observed feature vector, x_t.

A full derivation of conditional posterior distributions for each hidden variable in the model as needed by the Gibbs sampling procedure is beyond the scope of this paper but is provided in [13]. In practice, we reduce the inference load on the boundary variables, b_t, by exploiting acoustic cues in the feature space to eliminate the need for sampling on frames that are unlikely to be phonetic boundaries (i.e., $P(b_t = 0) = 1$). This is done by following the pre-segmentation method described in [29].

We should note that introducing boundary variables and allowing them to be sampled on or off during inference places this model in a unique space between more traditional HMM-based modeling and that of segment-based speech recognition [29]. While other methods use an initial segmentation to seed its HMM clusters, those segmentations tend to be fixed and subsequently discarded during later iterations of training in favor of the more traditional Viterbi decoding step [11], [21]. The model in [13] not only implements a form of duration modeling by forcing the 3-state HMM to represent the entire segment between two boundary variables $(b_l, b_r = 1)$, it also allows for its boundary variables to be sampled on and off $(b_t \in \{0, 1\})$. This allows the model to continuously refine both its segmentation and clustering at a more localized level without having to rely on HMMs to model the duration of an acoustic unit.

B. Parallelization

While Gibbs sampling is theoretically guaranteed to converge to the true posterior distribution of the hidden variables in [13], the process can be quite slow – the sampling of each variable in turn requires updates to all its dependent variables at each frame of audio. In an effort to scale from processing the relatively small TIMIT corpus [30] containing less than ten hours of speech to a corpus containing a few hundred hours of audio, we focused our efforts on parallelizing the sampling algorithm. The drawback of such parallelization is that the resulting algorithm, while computationally scalable, will only *approximate* Gibbs sampling [31]. There have been attempts to better understand these effects at a more theoretical level, but these initial studies have been restricted to simpler models [32]; the impact of more complex approaches has largely been observed empirically.

A traditional, serial Gibbs sampler samples from one conditional posterior distribution at a time and then updates its models accordingly. Our implementation resembles that of a blocked Gibbs sampler and conditions on all of the HMM and GMM parameters to sample, in parallel, a new set of alignments (i.e., per-frame segmentation boundaries and cluster assignments) for all utterances. Assuming our data is split into some arbitrary number of partitions, P, this allows for parallelization that can effectively decrease the required computation time by a factor of $\frac{1}{P}$. Given an entirely new set of alignments, we then accumulate statistics to update, in batch mode, our Dirichlet process counts, HMM transition probabilities, and GMM emission probabilities accordingly. Distributing the sampling process across a number of parallel workers before accumulating statistics globally is a technique that has been explored as a parallelized version of Latent Dirichlet Allocation (LDA) known as Approximate Distributed LDA (AD-LDA) [31]. Despite losing the traditional Gibbs sampling guarantee of converging to the true posterior distribution of the hidden variables, our implementation achieves performance on TIMIT that is comparable to that of [13] and even stronger empirical performance when given the opportunity to scale to large datasets.

C. Model Selection

Another difference between the model in [13] and our implementation is in the Dirichlet process (DP) mixture model. While the model allows for an infinite number of cluster labels in theory; practical implementations tend to over-initialize the number of possible mixtures and allow both the data and the DP concentration parameter, γ, to influence how many of those clusters actually retain probability mass. In our experiments, however, we found that our model would end up using all of the available mixtures, no matter how many we allowed for (up to 1000) or how small of a value we set for γ (as low as 0.001). This may have been a collective outcome of all our modifications; it may also indicate a lack of fit between our data and the model. Nevertheless, we found that the number of allowed clusters had a significant impact on both computational complexity and language recognition results; as such, we decided to fix the number of clusters at $|\mathcal{C}| = 100 << \infty$ and $\gamma = 1$, which achieves a balance between runtime and performance.

D. Other Modifications

We use the same pre-segmentation method used in [13] to obtain a set of candidate boundaries; this method essentially hypothesizes phonetic boundaries where the difference in spectral energy is large in magnitude. Originally built for a segment-based speech recognition system [29], [33], we further tuned this procedure to propose more candidate boundaries than usual, since the number of boundaries actually used (i.e., $b_t = 1$) will be a subset of those candidates. Lastly, while [13] imposed an equal prior probability on these candidate boundaries (i.e., $P(b_t = 1) = P(b_t = 0) = 0.5$), we found success in biasing the model towards keeping the boundary turned on with a prior of $P(b_t = 1) = 0.8$ and incorporating a localized post-processing step that merges consecutive segments if their respectively sampled cluster assignments are the same.

The model in [13] allows its HMMs to skip its middle and last states for segments shorter than three frames; our implementation does not allow for state-skipping and thus requires each acoustic unit to have a minimum duration of three frames. Our implementation also updates the GMM parameters in maximum likelihood fashion and increases the number of Gaussian mixtures it uses to model acoustic features at every pass through the data[2]; the formulation in [13] samples the emission probabilities of each HMM state using eight Gaussians with diagonal covariance matrices.

E. Unit Recognizer Training

The unit discovery process essentially produces an acoustic model consisting of HMM parameters for each individual acoustic unit (i.e., mono-unit), as well as a set of per-frame alignments from the training data indicating the Gaussian mixture, the HMM state, and the HMM cluster label that are associated with each acoustic feature vector. If we collapse these per-frame alignments into unit sequences, they can be used as transcripts to train a "unit recognizer" in the traditional way. This releases the model from the segment-based rigidity of boundary variables and lets boundaries be determined automatically via a forced alignment of the data. Recognizer training also allows for context-dependent modeling (i.e., "tri-units")—something our unit discovery method cannot do—which can ultimately provide us with per-frame alignment sequences in the form of senones [34].

In addition to showing the results of using per-frame unit sequences and per-frame HMM state sequences from our context-independent AUD, we will also show the results obtained at the speaker-independent (SI) and speaker-dependent (SD) stages of context-dependent unit recognizer training. For the SI stage, we build a context tree containing roughly 2500 senones, and for the SD stage, we use roughly 4500 senones and incorporate techniques for speaker adaptation such as MLLT (maximum likelihood linear transform), fMLLR (feature space maximum likelihood linear regression), and speaker adaptive training [26].[3] The exact number of senones obtained from the top-down, greedy splitting of the context tree is a function of the data and will differ between experiments; the resulting per-frame senone sequences are used as targets for training the DNN BN features.

III. The BN i-Vector System

In this section, we summarize the essential pieces of our BN i-vector language recognition system. To the extent possible,

[2] This is done according to the method used by default in Kaldi [26].

[3] These stages roughly follow the **tri2** and **tri4a** steps, respectively, in the `s5b` recipe of the Kaldi example for Switchboard I [26].

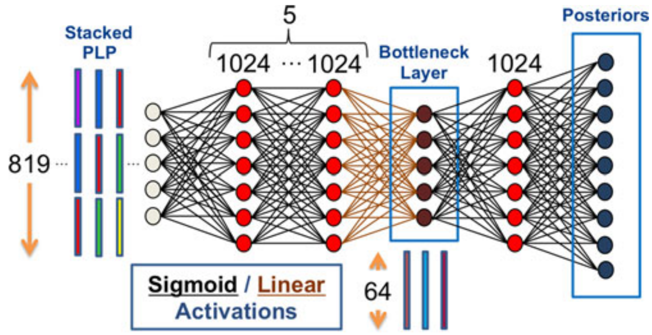


Fig. 3. The configuration of our proposed DNN. Its input is 819-dimensional vector of stacked PLP frames. The first five hidden layers contain 1024 nodes featuring sigmoid activations. This is followed by a 64-node BN layer that uses linear activations (from which we draw our BN features) and a final sigmoid layer with 1024 nodes. The number of output targets for which we obtain posteriors via a softmax depends on the result of the AUD step.

we followed the setup of the state-of-the-art LID system presented in [6]. As discussed in Section I, the overall process is summarized in Fig. 1.

A. DNN BN Features

A DNN classifier is essentially a multi-layer perceptron with more than two hidden layers that typically uses random initialization and stochastic gradient descent to initialize and optimize its weights [1]. To provide temporal context, the input to the DNN is typically a stacked set of spectral features extracted from short (20 ms) segments (frames) of speech. In our system, we compute 13 Gaussianized PLP coefficients as well as their first and second derivatives and then stack ± 10 frames of context around the current input frame to obtain a $(13 * 3) * (10 + 10 + 1) = 819$ dimensional input feature vector to the DNN. The (softmax) output of the DNN is trained using a cross-entropy cost function to predict the posterior probability of the target class for the current input frame; our experiments in Section IV will explore the use of unit cluster labels, c_t, hidden states, s_t, and senones as target classes.

We use this DNN as a means of extracting features for use by a secondary classifier (i.e., an i-vector system). This is accomplished by using the activation of one of the DNN's hidden layers as a feature vector. In particular, we optimize a dimension-reducing linear transformation as part of the DNN training that results in a special "bottleneck" layer with fewer nodes and, thus, a manageable dimensionality. The BN layer uses a linear activation and behaves very much like a LDA or PCA transformation on the activation of the previous layer [35], [36]. In addition to the previous work in [6], BN features also have been shown to work well for language recognition in [2], [5], [8], [9].

As illustrated in Fig. 3, all of our experiments utilize a common DNN structure containing seven hidden layers of 1024 nodes each with the exception of a BN sixth layer, which has 64 nodes instead. All hidden layers use a sigmoid activation function except for the fifth layer, which is linear [36]. As mentioned above, the input layer contains 819 input features covering 21 frames of context. In this setup, the only difference between experiments is the number of target classes, which is determined by the AUD system described in Section II.

B. i-Vector System and Scoring

While a detailed description of the i-vector system and theory is beyond the scope of this paper (but can be found in [37]), we provide a high-level overview of such a system built for language recognition (Fig. 1) and note that our framework closely follows that of [6], [12], [38]. In our experiments, the only difference between the various systems will be in the original acoustic/BN features used.

A test utterance whose language we hope to ascertain is first passed through a GMM-based speech activity detector, after which the detected speech is represented by a sequence of BN feature vectors as obtained from a DNN classifier explained above. From these features, we obtain the zeroth-order (counts) and first-order (means) sufficient statistics of the utterance from a Universal Background Model (UBM), which is a 2048-mixture GMM characterizing a speaker- and language-independent feature distribution. These statistics are then transformed into a raw i-vector of 600 dimensions using a total variability matrix, T [37]. We transform this raw i-vector using linear discriminant analysis (LDA) and within-class covariance normalization [12], [39], both of which are estimated a priori from the training data and their language labels, and finally length-normalize the result to obtain a test i-vector. We use the dot product to compute the similarity score between the test i-vector and each language-representing model i-vector. These scores are calibrated using a discriminative Gaussian backend described in [38], which is trained from a set of development data using both scores and utterance durations.

IV. Experiments

In this section, we first provide an overview of the data used in our experiments and present our initial results. Then we explore the use of score-level fusion and the incorporation of transcribed data on language recognition performance.

A. Corpora

Our experiments utilized three corpora in various ways. We evaluate all of our language recognition systems on the 2011 NIST *LRE11*, which covers 24 languages[4] coming from telephone and broadcast audio and has test durations of 3, 10, and 30 seconds [27]. The hyper-parameters for each of these systems—i.e., the UBM, the i-vector extractor, and the discriminative backend—are trained using the same training and development data from [38],[5] which we will refer to as LRE-train and LRE-dev, respectively. For our AUD on multilingual data, we used a subset of LRE-train consisting of 10 hours from each

[4] The LRE11 languages include Arabic-Iraqi, Arabic-Levantine, Arabic-Maghrebi, Arabic-MSA, Bengali, Czech, Dari, English-American, English-Indian, Farsi, Hindi, Lao, Mandarin, Pashto, Polish, Punjabi, Russian, Slovak, Spanish, Tamil, Thai, Turkish, Ukrainian, and Urdu.

[5] Some of the corpora represented include CallFriend, CallHome, Mixer, OHSU, and OGI-22, VOA, Radio Free Asia/Europe, GALE broadcasts, and Arabic corporal from the LDC and Appen [38].

of the 24 evaluation languages, yielding a 240-hour *LRE-subset* dataset. For proper comparison with previous work in [6], we also use a 100-hour subset of Switchboard I [10] as defined by the example system distributed with Kaldi [26], which we will abbreviate as *SWB*. Finally, while all of the experiments in this section report results on LRE11, we demonstrate the generalizability of our methods in Section V by applying them to the 2015 NIST LRE (*LRE15*) [40], [41], which features a modified evaluation protocol involving explicit language clusters.

B. Evaluation Metric

While more details regarding the evaluation metric can be found in [27], [38], we provide a brief overview in this section. The evaluation metrics put forth by NIST treat the language recognition problem as a series of verification tasks, in which the fundamental question is, "does test utterance τ belong to target language k?" In this way, we use false alarm rate, R_{FA}, and miss rate, R_{Miss}, in a way similar to the evaluation of speaker recognition. The only difference is that we express false alarm rate in the form of a language pair, $R_{\text{FA}}(k_T, k_N)$, which is the rate at which some specified non-target language, k_N, is mistaken for the target language, k_T. Under this paradigm, we obtain an average cost, C_{avg}, as

$$C_{\text{avg}} = \frac{1}{K}\left(C_{\text{Miss}}P_{\text{target}} \cdot \sum_{k_T} R_{\text{Miss}}(k_T) + \frac{1}{K-1}\left(C_{\text{FA}}P_{\text{non-target}} \cdot \sum_{k_T}\sum_{k_N \neq k_T} R_{\text{FA}}(k_T, k_N)\right)\right),$$

where K is the number of target languages (i.e., $K = 24$ for LRE11), and k_T and k_N denote target and non-target language, respectively. In our evaluations, the application-dependent costs for miss and false alarm errors, respectively, are set to be $C_{\text{Miss}} = C_{\text{FA}} = 1$, and the probability of target and non-target trials, respectively, are set at $P_{\text{target}} = P_{\text{non-target}} = 0.5$ [27], [40]. To make things easier to read, we will show our results as $C_{\text{avg}} \times 100$.

C. Spectral Feature Baseline

Following previous work, our baseline results come from an i-vector system built using MFCC-based spectral features. The baseline in [6], as well as in other work [12], [38], [42], [43], used Shifted Delta Cepstral (SDC) features in the conventional 7-1-3-7 scheme. The seven static cepstra are appended to the 49 SDC features to produce a 56-dimensional acoustic feature vector. A more detailed explanation on how the SDC are obtained can be found in [42]. Whereas the work in [38] included vocal tract length normalization and feature-domain nuisance attribute projection, these techniques are neither used in our work nor that of [6].

D. Transcribed SWB Benchmark

We use the results obtained in [6] as our supervised benchmark system. This system trains a DNN from 4,199 senone target labels generated at the *tri4a* step from the `s5b` recipe of the Kaldi example for Switchboard I [26], which we also adopted in Section II-E.

TABLE I
INITIAL LANGUAGE RECOGNITION RESULTS ON 30 SECOND TEST SEGMENTS OF LRE11; THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

	100 units	300 states	SI	SD
SWB (100 hrs)	9.10	7.36	6.29	5.89
LRE-subset (240 hrs)	9.02	6.67	5.65	**5.24**
Spectral Feature Baseline (Section IV-C)				5.29
Transcribed SWB Benchmark (Section IV-D)				**2.60**

The *SWB* row shows the results of a system built from acoustic units discovered on a 100-hour subset of Switchboard I (English), while the *LRE-subset* row corresponds to that of a system build from units discovered on 240 hours of multilingual data. The various columns show the results at different stages of unit discovery (unit cluster labels versus HMM hidden state labels) and unit recognizer training (speaker-independent and speaker-dependent). The bottom two rows show our baseline and benchmark results, respectively.

E. Initial Results

In our initial experiment, we fix the number of acoustic units at 100 and run AUD on SWB and LRE-subset. This results in per-frame unit sequences for the 100 units and corresponding 300 states (for each 3-state HMM), both of which can be used as targets for DNN training. As described in Section II-E, we also treat the resulting unit sequences as transcriptions and train an acoustic unit recognizer. We present our results obtained at two different stages of recognizer training: speaker-independent triphones (SI) and speaker-dependent (SD) modeling, which includes MLLT, fMLLR, and speaker adaptive training.

Table I presents our initial results, where for simplicity, we only show the detection cost on 30 second test segments of LRE11. In comparing between rows, we can see that running AUD on the multilingual LRE-subset is consistently better than running the unit discovery on the English-only SWB. This can be explained as either a result of domain adaptation to the multiple LRE11 languages or the effect of having 240 hours in the LRE-subset data versus 100 hours in SWB, or some combination of both. That said, the virtue of unsupervised methods is that they can be applied to the untranscribed multilingual data that matches the test domain; as such, subsequent results will be limited to units discovered on the LRE-subset.

Examining the columns from left to right, we can also see that each additional step of model refinement corresponds to additional improvements. Going from per-frame unit sequences (100) to context-independent HMM state sequences (300) yielded the most substantial gain; we note once again that both sets of sequences are solely the result of the AUD inference process (involving segment boundaries) and *not* a result of the unit recognizer training discussed in Section II-E.[6] During unit recognizer training, the SI step builds a phonetic

[6] Recall from Section II-E that the subsequent unit recognizer training ignores the original segment boundaries, b_t, from the AUD process and defines its own via the standard HMM training algorithms (forward-backward and Viterbi).

TABLE II
SCORE-LEVEL FUSION RESULTS ON LRE11 FOR VARIOUS TEST SEGMENT LENGTHS (30, 10, AND 3 SECONDS); THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

		30 sec	10 sec	3 sec
*	**Spectral Feature Baseline**	5.29	10.4	21.4
*	**AUD(LRE-subset), SD**	5.24	10.1	20.1
	Fusion of [*] above	**3.80**	**7.17**	**17.2**
	Transcribed SWB Benchmark	2.60	6.25	16.5

We fuse the Spectral Feature Baseline (Section IV-C) with the best-performing system from Table I, which was built on the LRE-subset data using AUD and speaker-dependent unit recognizer training (Section II-E). The Transcribed SWB Benchmark is discussed in Section IV-D.

TABLE III
SCORE-LEVEL FUSION RESULTS ON LRE11 FOR VARIOUS TEST SEGMENT LENGTHS (30, 10, AND 3 SECONDS); THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

		30 sec	10 sec	3 sec
	AUD(LRE-subset, MFCC), SD	5.24	10.1	20.1
**	**AUD(LRE-subset, SWB-BN), SI**	2.87	7.27	18.1
**	**Transcribed SWB Benchmark**	2.60	6.25	16.5
	Fusion of [**] above	**2.10**	**5.21**	**15.0**

The first row, **AUD(LRE-subset, MFCC), SD**, is the same result reported in Table II. As discussed in Section IV-G, the results in the second row, **AUD(LRE-subset, SWB-BN), SI**, incorporate transcribed SWB data into the AUD process in the form of BN features. Our best results are obtained by fusing this semi-supervised system with the Transcribed SWB Benchmark (Section IV-D).

decision tree in a data-driven, top-down, greedy fashion that yields ~ 2500 senones, while the SD step yields ~ 4500 senones and incorporates speaker adaptive training [26]. Potentially as a result of such increased temporal resolution, we are able to obtain results comparable to the acoustic baseline using SDC features (Section IV-C). However, such performance is still significantly worse than that of the transcribed SWB benchmark (Section IV-D).

F. Incorporating Fusion

Given the similar performance of both the unit discovery-based system and the spectral feature baseline, we present the result of a score-level system fusion (via multi-class logistic regression [38]) between the baseline and the best-performing system from Table I, which was built on the LRE-subset data using AUD and speaker-dependent recognizer training. The results inTable II suggest that the unit discovery-based system captures language-related information complementary to that of the spectral feature baseline. Fusing these two systems together yields a 27, 29, and 14% relative gain on 30-, 10-, and 3-second test segments, respectively, and significantly reduces the gap between the baseline and the transcribed SWB benchmark without using any transcribed data.

G. Incorporating Transcribed Data

The work presented thus far has focused on a fully unsupervised scenario that involves no transcribed data. Fusing an AUD-based system with a spectral feature baseline reduces the performance gap between the baseline and supervised system based on transcribed English. In practical scenarios, however, we will seldom be limited to situations in which we have absolutely no access to transcribed data or pronunciation dictionaries for any language. Instead, we are more likely to find ourselves in a situation where the linguistic knowledge we have at hand (e.g., American English) does not necessarily match the data we need to work with (e.g., the 23 other languages of LRE11). This situation was thoroughly explored in [6], [7] and is directly reflected in our transcribed SWB benchmark system (Section IV-D). In this section, we further investigate the effect of utilizing existing transcriptions, but strictly in the context of improving AUD for subsequent language recognition. We would like to see whether an improved representation of speech might result in the discovery of more salient acoustic units and thus improve language recognition performance.

To do so, we use the BN features obtained from the transcribed SWB benchmark system described in Section IV-D as the feature representation for AUD. That is, instead of using 39-dimensional MFCC features, we run the entire unit discovery and recognizer training process described in Section II using the 64-dimensional BN features extracted from the transcribed SWB DNN (Section IV-D). The resulting per-frame senone labels are then used as output targets to train (from scratch) a brand new DNN whose input layer is, as before, the original 819-dimensional stacked PLP features. In this way, the transcribed SWB BN features (SWB-BN) are used only as a pre-processing step for the unit discovery; thus, their impact manifests solely in the quality of the resulting per-frame senone labels.

Table III summarizes these results. The first row repeats the results from Tables I and II that ran AUD using MFCC features, while the second row displays the results of running AUD using SWB-BN features as described above. We can see the immediate impact of the improved feature representation and note that, unlike in the case of using MFCC features for AUD, SD training did not provide any improvement over SI modeling – this makes sense, since the SWB-BN features are trained using the speech of many speakers for the explicit purpose of discriminating between phonetic variabilities[7]—so we only show our SI results. But what is most important to realize is that these SWB-BN features are seen only by the unit discovery process to obtain the resulting per-frame senone sequences that we use as targets for subsequent DNN training. Everything else in our BN i-vector system, from stacked PLP coefficients as DNN inputs to i-vector extraction, remains exactly as described in Section III. Aside from the supervision involved in obtaining the SWB-BN features, the rest of the unit discovery system is still fully unsupervised. This clear difference in performance between the first and second rows of Table III demonstrates yet again the limitations of MFCC's as acoustic features.

Because we are now using transcribed SWB data to obtain our BN features, it is only fair to compare our results against those of the transcribed SWB benchmark. While this

[7] The work in [6] notes, however, that these same SWB-BN features can be effective for speaker recognition when used in tandem with spectral features.

TABLE IV
INDIVIDUAL AND SCORE-LEVEL FUSION RESULTS ON LRE11 FOR VARIOUS TEST SEGMENT LENGTHS (30, 10, AND 3 SECONDS); THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

		30 sec	10 sec	3 sec
1	**AUD(LRE-subset, SWB-BN), SI**	2.87	7.27	18.1
2	**SWB-ASR, Decode LRE-subset**	2.96	7.25	17.2
3	**SWB-BN DNN, Classify LRE-subset**	2.69	6.72	16.8
4	**Transcribed SWB Benchmark**	2.60	6.25	16.5
	Fusion: 1 + 4	**2.10**	5.21	15.0
	Fusion: 2 + 4	2.12	**5.07**	**14.2**
	Fusion: 3 + 4	2.16	5.11	14.9

All of the systems shown here use transcribed SWB in some way, and all but the benchmark system (Row 4) incorporate the use of LRE-subset data. *AUD* (Row 1) is the same result from Table III for a system that uses transcribed SWB data to obtain BN features for AUD on LRE-subset data. *SWB-ASR* (Row 2) uses a recognizer built from SWB to decode the LRE-subset data. *SWB-BN DNN* (Row 3) classifies each frame of the LRE-subset data using a DNN built from transcribed SWB data. Finally, our transcribed SWB benchmark result is shown again in Row 4.

benchmark is still the best individual system, its fusion with our (now semi-supervised) unit discovery-based system (using supervised SWB-BN features) yields relative gains of 19%, 16%, and 9% on 30-, 10-, and 3-second test segments, respectively. These results suggest that the information each of these two systems focuses on to make its classification decisions may be complementary.

We realize that there are a variety of other ways in which the transcribed SWB data can be utilized alongside the LRE-subset data in the form of unsupervised domain adaptation. For the sake of comparison, we explore a few of these methods but note that a thorough treatment of this problem is not the intended focus of this paper; our work simply aims to address the use of large-scale AUD as a tool to obtain features for language recognition. As a first experiment, we built and tuned an English recognizer from the transcribed SWB data (SWB-ASR), used the recognizer to decode the LRE-subset data, and built a new DNN from the corresponding per-frame senone sequences. For the next experiment, instead of a recognizer, we simply used the original SWB-BN DNN to classify each frame of the LRE-subset data. Those classification results were then used as (potentially noisy) targets to train a brand new DNN from scratch.[8] InTable IV, we can see that both experiments yielded individual and score-level fusion results that were similar to those obtained via our AUD BN i-vector system.

While the results shown in Table III demonstrate that even a little linguistic knowledge from just a single language (i.e., English) can have a huge impact in a multilingual setting (i.e., LRE11 performance), the results from Table IV further suggest that such supervision can be utilized in a variety of ways and still yield good results. We can also see that the original benchmark system (Row 4 of Table IV), which simply uses the original SWB-BN features for language recognition, continues to be the single best-performing system. This may imply that BN features are the most robust when trained using only labels obtained in a fully supervised fashion. We defer a more in-depth exploration of this phenomenon to future work.

V. DISCUSSION

So far, we have seen that a large-scale AUD system can yield a segmentation and clustering of untranscribed data that is useful for language recognition. And while our focus continues to be on a fully unsupervised approach, we have also seen that even a little bit of supervision can go a long way to improve results. In this section, we discuss some of the other approaches we tried that did not work as well as planned and then demonstrate the generalizability of our approach to the previously unseen LRE15 data.

A. Negative Results

In the development of our proposed system, we explored the use of various levels of supervision in initializing the AUD process. In one experiment, we initialized our HMMs with a set of transcription-derived alignments (i.e., SWB). In another, we initialized our HMMs at random, but at every iteration, we updated our models using statistics accumulated from both the newly-sampled alignments (on LRE-subset) *and* the transcription-derived alignments (from SWB). We also tried scaling the statistics from these respective alignments in various ways to adjust the amount our updated model relied on each one. Our hypothesis was that maintaining some level of supervision might help anchor an otherwise unsupervised process; however, this set of experiments yielded results that were no different from simply initializing the HMMs at random and accumulating statistics from just the newly-sampled alignments.

We also considered different ways to obtain our set of potential phonetic boundaries (i.e., $\{b_t\}$ from Section II). In addition to using the acoustic cues-based method from [29] with various parameter settings, we also experimented with the phone boundaries obtained from decoding the data using a speech recognizer trained on SWB. We found, however, that language recognition performance overall remained fairly stable across different segmentation methods, so long as the determined boundaries occurred at a reasonable frequency and, as discussed in Section II-D, the prior on the boundary variables is biased towards being turned on (i.e., $P(b_t = 1) = 0.8$). Because the unit recognizer training step subsequently redefines these initial boundaries, the quality of the discovered units seems to be most dependent on how well we can cluster the data given the feature representation (i.e., MFCC or SWB-BN).

B. The 2015 NIST Language Recognition Evaluation

Despite the amount of system development involved in obtaining the results described in Section IV, we demonstrate here that our proposed methods can generalize from LRE11 to a previously-unseen LRE. The 2015 NIST LRE encompasses 20 languages that can be grouped into six clusters[9] and, just like the LRE11, contains test durations of 3, 10, and 30 seconds [40]. Unlike previous evaluations, LRE15 focused on classifying target languages within the six language clusters. As such,

[8] Training a new DNN from scratch yielded better results than simply fine-tuning the original SWB-BN DNN.

[9] Arabic – Egyptian, Iraqi, Levantine, Maghrebi, Modern Standard; Chinese – Cantonese, Mandarin, Min, Wu; English – British, General American, Indian; French – West African, Haitian Creole; Iberian – Caribbean Spanish, European Spanish, Latin American Spanish, Brazilian Portuguese; Slavic – Polish, Russian.

TABLE V
RESULTS ON LRE15 BROKEN DOWN BY LANGUAGE CLUSTER—**AR**ABIC, **CH**INESE, **EN**GLISH, **IB**ERIAN, AND **SL**AVIC—THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

	Ar	Ch	En	Ib	Sl	Avg
*** Spectral Baseline**	26.6	23.4	16.9	23.4	11.4	20.3
*** AUD(MFCC), SD**	24.6	18.4	18.3	21.9	7.27	18.1
[*] above fused	**24.1**	**18.1**	**14.7**	**20.9**	**6.32**	**16.8**
**** AUD(SWB-BN), SI**	19.6	13.3	12.7	18.5	3.89	13.6
**** SWB-BN Benchmark**	19.6	13.1	11.2	18.4	3.27	13.1
[**] above fused	**18.6**	**11.8**	**10.3**	**17.1**	**2.89**	**12.1**

we present our results on each language cluster, with the exception of French, as well as an average over all clusters. It was determined during the post-evaluation workshop that the French language cluster data featured a systematic channel mismatch between the train and test segments that led to near-random classification performance for most submitted systems. Furthermore, it was noted that Haitian Creole has a range of spoken forms, with the more formal variety being more French-like and the informal variety much less so [41]. Addressing this issue is beyond the scope of this paper, so we omit these results; but for future work, it would be interesting to investigate more channel-robust (and speaker-independent) methods for unit discovery.

We used the development data provided by NIST: 141 hours of Arabic, 52 hours of Chinese, 65 hours of English, 4 hours of French, 23 hours of Iberian, and 30 hours of Slavic. This reflects the amount of speech in the data after speech activity detection. A more complete breakdown of the amount of speech provided for the languages within each cluster can be found in [41]. Despite the uneven distribution of these data across the various language clusters, we decided against selecting a more balanced subset and ran our initial experiments using all of it, including French.

The top three rows of Table V are analogous to the results shown in Table II, which respectively include baseline results using just spectral features, results from units discovered on MFCC features, and results from a fusion of the two. The lower three rows present results analogous to those shown in Table III, which include results from units discovered on SWB-BN features, benchmark results using just SWB-BN features, and results from a fusion of the two, respectively.

We can see in Table V that the same trends from our LRE11 results persist in LRE15, thus demonstrating the applicability of our approach to different languages. In fact, AUD using MFCC features does substantially better than the spectral feature baseline on all language clusters except for English. Furthermore, the performance of our AUD-based system using SWB-BN features is just about the same as that of the SWB-BN Benchmark, again with the exception of English.[10]

TABLE VI
RESULTS ON LRE15 BROKEN DOWN BY LANGUAGE CLUSTER—**AR**ABIC, **CH**INESE, **EN**GLISH, **IB**ERIAN, AND **SL**AVIC—THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

	# hrs	Ar	Ch	En	Ib	Sl	Avg
Ar	141	**25.1**	20.3	18.8	**22.0**	8.21	18.9
Ch	52	25.6	**19.8**	17.8	22.9	9.14	19.0
En	65	25.8	**19.8**	**15.9**	23.0	8.20	18.5
Ib	23	27.3	21.5	19.8	22.5	9.71	20.2
Sl	30	26.5	20.5	19.6	22.7	**7.63**	19.4
Fused	(311)	24.5	17.6	14.8	20.6	6.11	**16.7**
All Data	315	24.6	18.4	18.3	21.9	7.27	18.1

For each of these systems, we run AUD using only the data from the language cluster specified and build a language recognition system to classify languages from all five language clusters. The *Fused* system is a score-level fusion of the five language cluster-specific systems shown; the results of a system obtained using *All Data* (including French) to discover acoustic units is the same as the second row in Table V.

C. Exploring Language Specificity

Because the LRE15 explicitly focuses on distinct language clusters, we also explore the impact of language-specific perspectives in our unit discovery. Each row ofTable VI shows the result of building a language recognition system via unit discovery (on MFCC features) on just the specified language cluster.[11] In each column, we highlight the best result obtained on the corresponding language cluster. This yields a fairly strong diagonal, where the only off-diagonal element is likely due to the lack of Iberian data relative to the amount of Arabic data. Otherwise, our results seem to confirm the notion that learning units on a particular family of related languages does indeed improve recognition performance for that specific language cluster, which may not be terribly profound but serves as further evidence that our AUD process captures language-specific information. Finally, fusing together the five language cluster-specific systems also yields a stronger result on each language cluster than simply discovering acoustic units from all languages (including French) pooled together.

A natural extension of these results would be a set of experiments that separate the impact of data amount from that of language-specificity. In particular,Table VII shows the results of running unit discovery on the same amount of data from each language cluster and building respective language recognition systems for each one. Iberian is our most data-limited language cluster, so we used all of its data and randomly selected a 23 hour subset of data from every other language cluster for AUD. Upon highlighting the corresponding row that obtains the best result for each column, we see a diagonal that is not confounded by an imbalance of training data between language clusters. Compared to the results shown in Table VI, we can see that the decrease in data for AUD on each language cluster seems to decrease language recognition performance on average, but not all language clusters are affected in the same way. In particular, all of the results on the English language cluster actually improve with the decrease in the amount of provided data. Table V

[10] We should also note that in our actual submission to the LRE15, the two best performing individual systems were the SWB-BN Benchmark with a slightly different DNN configuration (i.e., 80 nodes at the BN layer) and our AUD-based system as described here [41].

[11] For reasons described in the previous section, we do not build a language cluster-specific system using the French data.

TABLE VII
RESULTS ON LRE15 BROKEN DOWN BY LANGUAGE CLUSTER—**AR**ABIC, **CH**INESE, **EN**GLISH, **IB**ERIAN, AND **SL**AVIC—THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

	# hrs	Ar	Ch	En	Ib	Sl	Avg
Ar	23	**25.8**	21.4	18.6	22.9	8.84	19.5
Ch	23	26.7	**21.0**	17.5	22.9	10.2	19.7
En	23	26.5	21.8	**15.7**	23.0	9.37	19.3
Ib	23	27.3	21.5	19.8	**22.5**	9.71	20.2
Sl	23	26.1	21.2	19.1	**22.5**	**8.69**	19.5
Fused	(115)	**24.9**	18.4	**14.2**	**20.7**	**7.00**	**17.0**
All 5 Subsets	115	25.3	**18.2**	16.4	22.0	7.89	18.0

For each of these systems, we run AUD using only a 23 hour subset of the data from the language cluster specified and build a language recognition system to classify languages from all five language clusters. The *Fused* system is a score-level fusion of the five language cluster-specific systems shown, and the results of a system obtained using *All 5 Subsets* (excluding French) to discover one set of acoustic units are shown in the last row.

TABLE VIII
RESULTS ON LRE15 BROKEN DOWN BY LANGUAGE CLUSTER—**AR**ABIC, **CH**INESE, **EN**GLISH, **IB**ERIAN, AND **SL**AVIC—THE NUMBERS SHOWN ARE THE AVERAGE DETECTION COSTS $C_{\text{AVG}} \times 100$

	# hrs	Ar	Ch	En	Ib	Sl	Avg
Ar	23	**20.9**	16.0	15.2	20.3	6.39	15.8
Ch	23	22.1	16.1	15.2	20.3	5.72	15.9
En	23	21.6	15.4	**12.8**	19.2	5.84	15.0
Ib	23	21.4	**15.3**	15.5	**19.1**	5.40	15.3
Sl	23	21.3	16.0	15.6	20.8	**4.66**	15.7
Fused	(115)	**19.5**	**12.9**	**11.2**	**17.6**	**3.53**	**12.9**
All 5 Subsets	115	20.2	14.3	13.1	18.3	3.94	14.0
All Data	315	19.6	13.3	12.7	18.5	3.89	13.6
SWB-BN Benchmark		19.6	13.1	**11.2**	18.4	**3.27**	13.1

For each of these systems, we represent the audio using SWB-BN features and run AUD using a 23 hour subset of the data from the language cluster specified. Using these discovered acoustic units, we build a language recognition system to classify languages from all five language clusters.

also reflects this anomaly: the spectral baseline significantly outperforms the unit discovery method (16.9 versus 18.3) on the English language cluster. Our future investigations will explore the causes of such systematic discrepancies in our results on the English language cluster.

As expected, score-level fusion of the five separate language cluster-specific systems provides the best result on each individual language cluster and overall. The fused system (Fused, 115 hours) also achieves better performance on average and on each individual language cluster except Chinese when compared to the experiment shown in the last row of Table VII in which we pool the 23 hour subsets together and discover a single set of units on all of the languages combined (All 5 Subsets, 115 hours). Lastly, comparing the (Fused, 115) and (All 5 Subsets, 115) results in Table VII with the (Fused, 311) and (All Data, 315) results in Table VI, we can see that the results are actually quite similar despite a three-fold difference in the amount of data used to for AUD. This seems to suggest that the amount of data may not be the primary factor inhibiting performance. We should also note, however, that a three-fold increase in the amount of data does not necessarily imply a three-fold increase in computation time. In the "embarrassingly parallel" paradigm outlined in Section II-B, an increase in data can be remedied with a sub-linear increase in computational complexity by simply splitting the data into additional partitions.

All of the results from Tables VI and VII are obtained via AUD on MFCC features and, despite significantly improving upon the spectral feature baseline result in Table V, are still far from the respective performances of the SWB-BN benchmark and AUD-based system that uses SWB-BN features. Our original intention was to ascertain the true effect of language specificity at the spectral feature level; as such, we chose not to use these English-inspired features in our initial investigation. But for completeness, Table VIII presents the result of using SWB-BN features for AUD on a per-language cluster basis, and we can see the corresponding performance improvements. In addition to being better than systems in which all languages are pooled together for AUD, our fusion of language-specific systems does even a bit better than the SWB-BN Benchmark on average.

Nevertheless, our experiment results continue to motivate the need for an improved feature representation for all languages. Our results in Table VIII were obtained using transcribed English and did fairly well, but there is a lot of room for further improvement. For instance, appropriate features for English may be insufficient for other language clusters. The tonal nature of Chinese, in particular, is completely ignored in the standard MFCC-based representation used for English. As we look ahead to future work, one experiment that would help ascertain the generalizability of our methods is to use a more appropriate feature representation for Chinese, learn acoustic units on those, and evaluate the new system.

VI. CONCLUSION

In this paper, we explored the application of large-scale AUD as a tool to obtain features for language recognition. To do so, we implemented a parallelized version of a Bayesian nonparametric model from previous work and used it to learn acoustic units from hundreds of hours of multilingual data. We use these per-frame sequences of units (or states or senones) as targets to train a DNN that, given stacked spectral features as input, provides a BN feature representation that can be used for i-vector language recognition. We found that a score-level fusion with a baseline system built from acoustic features yields substantial gains and significantly closes the gap between the baseline and a benchmark system built using transcribed English, suggesting that discovered acoustic units may be complementary to spectral features. Subsequently, we also found that an improved representation of speech (i.e., supervised BN features) as input to our AUD system can yield substantial performance gains, thus motivating the need for a better understanding of the speech signal. We validated the generalizability of our proposed approach by presenting results that exhibit similar trends on the LRE's from both 2011 and 2015. Finally, we demonstrated on LRE15 the continued importance of language specificity for unit discovery.

As we have commented throughout this paper, there exist many avenues for future work. Apart from a cursory visual inspection, we have not done much to measure, using existing metrics [44], the quality of the discovered acoustic units using such a large set of multilingual data. We can also update our DNN architecture to account for language specificity at the output layer; for example, each target class could be a language-specific senone where, during training, we compute our softmax only for the senones pertinent to the current language [45], [46]. Continuing to explore language specificity is certainly ripe with possibilities, especially in the context of LRE15 and our observations with the English and French language clusters. Furthermore, another easy extension mentioned in the previous section would be to consider the use of different acoustic features for different languages—running unit discovery using MFCCs is far less meaningful on tonal languages such as Chinese.

The notion of language-specific perspectives can be seen as a form of weak supervision, and we have also seen that a little supervision of any form and from any language can be extremely useful. To this end, there are many ways to incorporate this that can still be explored, including pairwise constraints on acoustic sequences (e.g., examples of the same word) [47]–[49]. And lastly, unsupervised AUD can also be seen as a tool to ascertain the phonotactics of a language; it would be interesting to extend our work to more recent phonotactic approaches to language recognition using our discovered units [50]–[52]. For example, we can use our language-specific acoustic unit models to tokenize the multilingual data and build a system analogous to the approach known as parallel phoneme recognition followed by language modeling (PPR-LM) [22]–[25]. Some of our initial work on this front can be found in Section 5.6 of [53].

Acknowledgements

The authors would like to thank Fred Richardson for sharing his DNN expertise and implementation, as well as Douglas Reynolds and Ekapol Chuangsuwanich for their guidance and extremely helpful insights throughout the development of this work.

References

[1] G. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, “i-vector representation based on bottleneck features for language identification,” *Electron. Lett.*, vol. 49, no. 24, pp. 1569–1570, 2013.

[3] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, “Spoken language recognition based on senone posteriors,” in *Proc. Interspeech*, 2014, pp. 2150–2154.

[4] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2014, pp. 695–1699.

[5] P. Matejka *et al.*, “Neural network bottleneck features for language identification,” in *Proc. Odyssey*, 2014, pp. 299–304.

[6] F. Richardson, D. Reynolds, and N. Dehak, “Deep neural network approaches to speaker and language recognition,” *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1671–1675, Oct. 2015.

[7] F. Richardson, D. Reynolds, and N. Dehak, “A unified deep neural network for speaker and language recognition,” in *Proc. Interspeech*, 2015, pp. 1146–1150.

[8] R. Fer, P. Matejka, F. Grezl, O. Plchot, and J. Cernocky, “Multilingual bottleneck features for language recognition,” in *Proc. Interspeech*, 2015, pp. 389–393.

[9] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, “Study of senone-based deep neural network approaches for spoken language recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 1, pp. 105–116, Jan. 2016.

[10] J. J. Godfrey, E. C. Holliman, and J. McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 1992, pp. 517–520.

[11] M.-H. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, “Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery,” *Comput. Speech Lang.*, vol. 28, no. 1, pp. 210–223, 2014.

[12] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *Proc. Interspeech*, 2011, pp. 857–860.

[13] C. Lee and J. Glass, “A nonparametric Bayesian approach to acoustic model discovery,” in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, 2012, pp. 40–49 .

[14] A. Jansen *et al.*, “A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2013, pp. 8111–8115.

[15] A. Jansen, K. Church, and H. Hermansky, “Towards spoken term discovery at scale with zero resources,” in *Proc. Interspeech*, 2010, pp. 1676–1679.

[16] A. Jansen and K. Church, “Towards unsupervised training of speaker independent acoustic models,” in *Proc. Interspeech*, 2011, pp. 1693–1692.

[17] R. Singh, B. Raj, and R. M. Stern, “Automatic generation of subword units for speech recognition systems,” *IEEE Trans. Speech Audio Process.*, vol. 10, no. 2, pp. 89–99, Feb. 2002.

[18] S. Chaudhuri, M. Harvilla, and B. Raj, “Unsupervised learning of acoustic unit descriptors for audio content representation and classification,” in *Proc. Interspeech*, 2011, pp. 2265–2268.

[19] G. Chollet, J. Cernocky, A. Constantinescu, S. Deligne, and F. Bimbot, “Toward ALISP: A proposal for Automatic Language Independent Speech Processing,” in *Computational Models of Speech Pattern Processing*. Berlin, Germany: Springer, 1999, pp. 375–388.

[20] D. Petrovska-Delacretaz, M. Abalo, A. E. Hannani, and G. Chollet, “Data-driven speech segmentation for language identification and speaker verification,” in *Proc. Non Linear Speech Process.*, 2003.

[21] H. Khemiri, “Unified data-driven approach for audio indexing, retrieval, and recognition,” Ph.D. dissertation, TELECOM ParisTech, Paris, France, 2013.

[22] M. Zissman, “Comparison of Four Approaches to Automatic Language Identification of Telephone Speech,” *IEEE Trans. Speech Audio Process.*, vol. 4, no. 1, pp. 31–44, Jan. 1996.

[23] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, “Phonotactic language recognition using high quality phoneme recognition,” in *Proc. Interspeech*, 2005, pp. 2833–2836.

[24] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. Deller, Jr., “Language identification using Gaussian mixture model tokenization,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2002, pp. I-757–I-760.

[25] H. Li, B. Ma, and C. Lee, “A vector space modeling approach to spoken language identification,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 271–284, Jan. 2007.

[26] D. Povey *et al.*, “The kaldi speech recognition toolkit,” in *Proc. IEEE Automat. Speech Recog. Understanding Workshop*, 2011.

[27] NIST, “The 2011 NIST language recognition evaluation plan (LRE11),” 2011. [Online] Available: http://www.nist.gov/itl/iad/mig/lre11.cfm

[28] A. Garcia and H. Gish, “Keyword spotting of arbitrary words using minimal speech resources,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, 2006, pp. 949–952.

[29] J. Glass, “A probabilistic framework for segment-based speech recognition,” *Comput. Speech Lang.*, vol. 17, pp. 137–152, 2003.

[30] J. S. Garofolo *et al.*, “TIMIT acoustic-phonetic continuous speech corpus LDC93S1,” Philadelphia: Linguistic Data Consortium, 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1

[31] A. Ihler and D. Newman, “Understanding errors in approximate distributed latent Dirichlet allocation,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 952–960, May 2012.

[32] M. J. Johnson, J. Saunderson, and A. S. Willsky, “Analyzing hogwild parallel Gaussian Gibbs sampling,” in *Proc. Adv. Neural Inform. Process. Syst.*, 2013, pp. 2715–2723.

[33] J. R. Glass and V. W. Zue, “Multi-level acoustic segmentation of continuous speech,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 1988, pp. 429–432.

[34] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. Workshop Human Lang. Technol.*, 1994, pp. 307–312.

[35] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2013, pp. 6655–6659.

[36] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2014, pp. 185–189.

[37] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, Jul. 2010.

[38] E. Singer *et al.*, "The MITLL NIST LRE 2011 Language recognition system," in *Proc. Odyssey*, 2012, pp. 209–215.

[39] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp. 1471–1474.

[40] NIST, "The 2015 NIST language recognition evaluation plan (LRE15)," 2015. [Online]. Available: http://www.nist.gov/itl/iad/mig/upload/LRE15_EvalPlan_v23.pdf

[41] P. Torres-Carrasquillo *et al.*, "The MITLL NIST LRE 2015 language recognition system," in *Proc. Odyssey*, 2016, pp. 196–203.

[42] P. A. Torres-Carrasquillo *et al.*, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. Int. Conf. Spoken Lang. Process.*, 2002, pp. 89–92.

[43] B. Bielefeld, "Language identification using shifted delta cepstrum," in *Proc. 14th Annu. Speech Res. Symp.*, 1994.

[44] M. Versteegh *et al.*, "The zero resource speech challenge 2015," in *Proc. Interspeech*, 2015, pp. 3169–3173.

[45] F. Grezl, M. Karafiat, and K. Vesely, "Adaptation of multilingual stacked bottleneck neural network structure for new language," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2014, pp. 7654–7658.

[46] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Language ID-based training of multilingual stacked bottleneck features," in *Proc. Interspeech*, 2014, pp. 1–5.

[47] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2013, pp. 8091–8095.

[48] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2015, pp. 5818–5822.

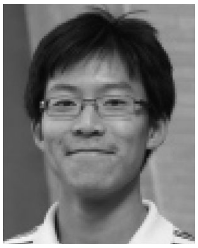
[49] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2016, pp. 4950–4954.

[50] T. Mikolov, O. Plchot, O. Glembek, P. Matejka, L. Burget, and J. Cernocky, "PCA-based feature extraction for phonotactic language recognition," in *Proc. Odyssey*, 2010.

[51] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "iVector approach to phonotactic language recognition," in *Proc. Interspeech*, 2011, pp. 2913–2916.

[52] M. Soufifar, L. Burget, O. Plchot, S. Cumani, and J. Cernocky, "Regularized subspace n-gram model for phonotactic ivector extraction," in *Proc. Interspeech*, 2013, pp. 74–78.

[53] S. H. Shum, "Overcoming resource limitations in the processing of unlimited speech: Applications to speaker and langauge recognition," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, Jun. 2016.



Stephen H. Shum received the B.S. degree in electrical engineering and computer science (EECS) from the University of California, Berkeley, CA, USA, in 2009 before joining the Spoken Language Systems Group at the MIT Computer Science and Artificial Intelligence Laboratory, where he received the S.M. degree in 2011. He received the Ph.D. degree in EECS from the Massachusetts Institute of Technology, CA, USA. He was awarded the William A. Martin Thesis Award for his work on speaker diarization. Since then, Stephen has dabbled in the likes of speaker and language recognition while maintaining interest in a variety of topics, including semi-supervised learning, computational auditory scene analysis, and large-scale clustering of audio corpora.

David F. Harwath received the B.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2010 and is currently working toward the Ph.D. degree in Spoken Language Systems Group at the MIT Computer Science and Artificial Intelligence Laboratory. His research interests range from speech and audio signal processing to natural language understanding to computer vision. He has published papers on language identification, latent topic modeling and text summarization, pronunciation modeling for speech recognition, and unsupervised analysis of patterns in speech audio. His current research interests includeunsupervised, semantic joint modeling of speech audio and visual images.

Najim Dehak received the Ph.D. from the School of Advanced Technology, Montreal, in 2009. During his Ph.D. studies he worked with the Computer Research Institute of Montreal, Canada. He is well known as a leading developer of the i-vector representation for speaker recognition. He first introduced this method, which has become the state-of-the-art in this field, during the 2008 Center for Language and Speech Processing Summer Workshop at Johns Hopkins University. This approach has become one of most known speech representations in the entire speech community.

He is currently a Faculty Member of the Department of Electrical & Computer Engineering at Johns Hopkins University. Prior to joining Johns Hopkins, he was a Research Scientist in the Spoken Language Systems Group at the MIT Computer Science and Artificial Intelligence Laboratory. His research interests include machine learning approaches applied to speech processing, audio classification, and health applications. He is a member of the IEEE Speech and Language Technical Committee.

James R. Glass received the B.Eng. degree in electrical engineering from Carleton University, Ottawa, ON, Canada, in 1982, and the S.M. and Ph.D. degrees in electrical engineering and computer science at Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1985, and 1988, respectively.

He is a Senior Research Scientist at MIT where he leads the Spoken Language Systems Group in the Computer Science and Artificial Intelligence Laboratory. He is also a member of the Harvard-MIT Health Sciences and Technology Faculty. Since obtaining his doctorate at MIT in Electrical Engineering and Computer Science, his research has focused on automatic speech recognition, unsupervised speech processing, and spoken language understanding. He is an Associate Editor for Computer, Speech, and Language, and is an IEEE Fellow, and a Fellow of the International Speech Communication Association.