Study of Senone-Based Deep Neural Network Approaches for Spoken Language Recognition

Luciana Ferrer, Yun Lei, Mitchell McLaren, and Nicolas Scheffer

***Abstract*—This paper compares different approaches for using deep neural networks (DNNs) trained to predict senone posteriors for the task of spoken language recognition (SLR). These approaches have recently been found to outperform various baseline systems on different datasets, but they have not yet been compared to each other or to a common baseline. Two of these approaches use the DNNs to generate feature vectors which are then processed in different ways to predict the score of each language given a test sample. The features are extracted either from a bottleneck layer in the DNN or from the output layer. In the third approach, the standard i-vector extraction procedure is modified to use the senones as classes and the DNN to predict the zero-th order statistics. We compare these three approaches and conclude that the approach based on bottleneck features followed by i-vector modeling outperform the other two approaches. We also show that score-level fusion of some of these approaches leads to gains over using a single approach for short-duration test samples. Finally, we demonstrate that fusing systems that use DNNs trained with several languages leads to improvements in performance over the best single system, and we propose an adaptation procedure for DNNs trained with languages with less available data. Overall, we show improvements between 40% and 70% relative to a state-of-the-art Gaussian mixture model (GMM) i-vector system on test durations from 3 seconds to 120 seconds on two significantly different tasks: the NIST 2009 language recognition evaluation task and the DARPA RATS language identification task.**

***Index Terms*—Spoken Language Recognition, Deep Neural Networks, Senones**

I. Introduction

In recent years, speech-processing researchers have started exploring the use of deep neural networks (DNNs) for several different tasks, including automatic speech recognition (ASR), speaker recognition and spoken language recognition (SLR). Perhaps most prominent in the literature is the successful application of DNNs to ASR, replacing Gaussian mixtures for modeling the acoustic features (for example, see [1], [2]). In SLR, researchers have explored several different strategies for using DNNs. As we describe in this introduction, the most successful of these approaches are those based on senone-driven DNNs. The goals of this work are (1) to compare these approaches with each other and with a common baseline in a unified framework, and (2) to propose simple improvements for these techniques.

New SLR approaches are generally compared with what is currently considered the state of the art in SLR. This system consists of extraction of shifted delta cepstrum (SDC) features followed by i-vector modeling, an approach proposed for speaker recognition in [3] and first applied to SLR in [4] and [5]. The i-vectors for the test utterances are then modeled by standard techniques like the Gaussian backend (GB), neural network, or logistic regression [4] to produce scores for each target language.

Three main approaches that use senone-driven DNNs for SLR can be found in recent literature. The first approach was proposed by our group in [6], [7] and, in parallel, in [8] for speaker recognition. In this method, the standard i-vector technique [3] is modified to use senones as classes instead of the Gaussians defined by a Gaussian mixture model (GMM)-based universal background model (UBM). The zero-th order statistics needed for i-vector extraction are estimated using a DNN. This approach led to impressive relative gains of more than 30% on the SLR data from the DARPA RATS program [7]. We call this method the **DNN/iv** modeling approach. To our knowledge, results from this approach on the more standard language recognition data from the NIST evaluations have not yet been published.

The second approach, which we call **DNN/post**, was proposed more recently by our group. It uses the DNN output layer to create features for language recognition. The posteriors for each senone at each frame are processed to generate a single vector per utterance, which is then modeled with standard backend techniques ([7], [9]). Again, this approach had only been tested on RATS data until this study.

A third approach, first proposed for SLR in [10], is to train a DNN with a bottleneck architecture and then use the outputs of the bottleneck layer as features within an i-vector framework. In the aforementioned work, the DNN is trained to predict the phone posteriors using 1000 hours of Mandarin data. The authors show impressive gains with respect to a state-of-the-art SDC i-vector system on the NIST 2009 Language Recognition Evaluation (LRE) data. A more recent publication from the same author [11] explores the different parameters of the proposed system, including a fusion of the Mandarin-based DNN system with an English-based DNN system, showing gains from the fusion of between 7% and 18% for the different

L. Ferrer is with the Computer Science Department, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina, and with CONICET, Argentina.

M. McLaren is with the Speech Technology and Research Laboratory, SRI International, California, USA.

Y. Lei and N. Scheffer were also with SRI while their work on this project was done. They are now currently with Facebook, Inc., USA.

This material is based on work partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract D10PC20024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA or its contracting agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. "A" (Approved for Public Release, Distribution Unlimited)

Manuscript received December 29, 2014; revised June 23, 2015; accepted October 13, 2015.

test conditions.

This approach was also proposed, almost in parallel, in [12]. In this case, the authors test on the RATS program data, training the DNNs on two languages, Farsi and Levantine Arabic, to predict posteriors for both context-independent and context-dependent phones. They show significant gains from using context-dependent phones and from fusing systems that use the two different DNNs. They also propose an approach where two DNNs are stacked: the bottleneck features from the first DNN over a context of 10 frames are used as input for a second DNN with the same architecture as the first one. They show relative gains of 4% and 10% in the 3- and 10-second conditions, respectively, from the stacked approach compared with the single-DNN approach. In this work, we focus on the single-DNN approach for simplicity. We call the features extracted from the bottleneck layer deep bottleneck features or **DBFs**.

All these approaches have shown significant gains over well-established state-of-the-art systems. Unfortunately, results from the various approaches cannot be compared across papers for a few reasons. First, all these approaches rely on using DNNs trained to predict phone-related units. Different groups are using different data to train these DNNs, making the results incomparable. Second, the features that are used as input to the DNNs are also different, adding another variable that is likely to affect results. Third, the baseline results are different across groups, making the relative gains less meaningful when compared across papers. Finally, the two tasks that are mostly chosen as test data, RATS SLR and LRE09 are quite different from each other. While the RATS data is extremely noisy and distorted, the LRE data is much cleaner and includes only telephone-bandwidth data. Further, the RATS SLR task comprises only 5 target languages, while the LRE09 task includes 23 languages with an extremely unbalanced number of samples per language in the training data.

We believe it is important to test whether the gains obtained on one of the tasks also hold on the other task, and to compare the approaches within each task in a single development environment where most variables are kept unchanged. Fusion experiments across approaches are also missing from current literature. We believe we are in a great position to perform these comparisons, given that our DNN-based results have proven to be outstanding for both SLR and speaker recognition tasks. A number of techniques highly related to the ones described above will not be included in this study. Below, we briefly describe them and justify our decision to leave them out of this work.

The work in [13] proposed directly modeling language posteriors with a DNN. In this approach, perceptual linear prediction (PLP) features are used as input to a DNN trained to predict the target languages, and, optionally, an out-of-set class. The authors reported, during the oral presentation of their work, that this approach is inferior to using bottleneck features within a standard i-vector modeling framework. For this reason, we do not include results for this approach in this paper.

Another approach, highly related to the DNN/post and DBF approaches above, was proposed earlier in [14]. In this work, a NN is used to predict phone-level posteriors (around 50 units, depending on the language) at a frame level; these posteriors are then converted into log-likelihood ratios (LLRs). These LLRs are used to extract i-vectors, which are then modeled using a Gaussian backend. This approach is similar to DNN/post in that it uses the output layer of a NN, and similar to DBF in that it models features extracted from phonetically-driven DNNs using an i-vector approach. Results are approximately 10% better than the baseline SDC i-vector system on the 30-second test condition of LRE09, a relative gain significantly smaller than what we obtain on the same task for the DNN/post and DBF systems. Further, as shown in [12], using a DNN trained to predict context-dependent phones like the senones we use in this work is clearly better than using a DNN trained to predict context-independent phones. For these reasons, we do not include a comparison to this system here.

Finally, the DNN/post approach, which models the senone posteriors over the whole utterance, can be considered a phonotactic approach. Phonotactic approaches attempt to model the permissible combinations of phones and their frequencies in the languages of interest. Standard phonotactic approaches involve collecting the probabilities for phone sequences as a representation of the signal by using the output of one or several open-phone loop recognizers ([15], [16], [17]). Language models or support vector machines are then used to generate the final scores. Another phonotactic approach uses the phoneme posteriogram counts from the phone recognizer to create bigram conditional probabilities, which are then used to create features for SLR (e.g., [18]). These phonotactic approaches work with a relatively small set of units (usually approximately 50) representing the individual phones of the language being modeled. Information about the frequency of different phone sequences is collected through n-gram generation. The best phonotactic approaches perform somewhat worse than the current state-of-the-art systems. The DNN/post approach, on the other hand, clearly outperforms current state-of-the-art systems and is much simpler than those approaches in that it does not require explicitly generating n-grams and the subsequent selection usually needed for computational reasons. Hence, we will not include a comparison to these older phonotactic approaches.

The current work presents a careful comparison and analysis of the selected approaches described above and their fusion. We present results on the LRE09 task, selected for being a standard in the SLR literature against which most groups will be able to compare. For this task, for which the training data is highly language-imbalanced, we introduce a weighted Gaussian backend that results in consistent gains over the standard Gaussian backend. We compare and fuse systems from the three approaches using DNNs trained with data from four languages: English, Spanish, Mandarin Chinese, and colloquial Egyptian Arabic. For the last three languages, which have much less data available than English, we propose an adaptation technique for the DNN that results in large gains in SLR performance. Finally, we also show results on the RATS task, demonstrating that the main conclusions carry over across these two very different datasets. Both the weighted Gaussian backend and the DNN-adaptation technique applied to SLR

tasks are novel contributions of this paper.

The rest of the paper is organized as follows. Section II describes the technology underlying the DNN-based approaches that will be explored in this paper. Sections III and IV describe the setup and results obtained for the LRE09 and RATS SLR tasks, respectively. We present our conclusions based on these results in Section V.

II. Background and System Descriptions

With the exception of the baseline, all approaches compared in this paper rely on DNNs trained to predict the posterior probability of senones at the frame level. The following sections first explain what senones are, how DNNs are used to estimate their posterior probabilities given a set of features, and how these DNNs are trained. We then describe the features that can be derived from these DNNs and the two i-vector approaches used in this work: the standard one and the one based on DNN posteriors. Finally, we present the backend and calibration techniques used for all systems, and we give a summary of the architecture of each of the compared systems.

A. Senone-Driven DNNs

This section defines the concept of senones and explains how their posteriors are calculated by using a DNN and how these DNNs are trained.

1) Senone Definition: Senones are defined as states within context-dependent phones. Senones are the unit for which observation probabilities are computed during ASR. The pronunciations of all words are represented by a sequence of senones $\mathcal{Q}$. In general, the senone set $\mathcal{Q}$ is automatically defined by a decision tree [19]. At every node, a question is asked from a predefined set that includes questions about the left and right context, the central phone, and the state number. An example question could be: "Is the phone to the left of this central phone a nasal?" The decision tree is grown in a greedy, top-down manner by selecting at each node the question that gives the largest likelihood increase, assuming that the data on each side of the split can be modeled by a single Gaussian. The leaves of the decision tree are then taken as the final set of senones. An example of a senone could be the first state of all triphones where the central phone is /iy/, the right context is a nasal, and the left context is /t/.

2) Senone Posterior Estimation Using a DNN: Traditionally, a GMM was used to model the likelihood of the senones $p(x|q)$ for ASR. Recent studies have shown that DNNs can be successfully used to estimate the senone posteriors $p(q|x)$ at the frame level, which are then converted into likelihoods using Bayes rule. This practice has significantly improved ASR performance relative to traditional GMM-based systems ([1], [2]).

As described in the introduction, senone-driven DNNs have recently also been applied to the SLR task using different approaches. Different input features have been used in these studies. In [10], 39-dimensional MFCC features plus 4 pitch features over 10 frames are concatenated and used as input to the DNN. In [12], frequency-domain linear prediction features are used as input to the DNN. The goal of this paper is not to optimize the input features to the DNN, but rather to compare SLR methods, given a certain DNN architecture and input feature set. For this reason, we simply use the standard features used in ASR for senone posterior estimation by DNNs: the log mel-filterbank coefficients over some context given by a few frames around the target frame.

In this paper, we consider two approaches for senone posterior estimation: multilayer perceptron and convolutional neural network. Nevertheless, one could consider using other deep learning techniques that have been shown to give excellent performance in ASR, such as deep convex networks (DCNs, [20], [21]) and long short-term memory recurrent neural networks (LSTM RNNs, [22]). Systems based on posterior probabilities from the last layer would remain unchanged when using these alternative approaches for the estimation of these posteriors. Extraction of bottleneck features for these cases, though, would be a research question.

3) DNN Architectures: We use two DNN architectures in this study. The **full** architecture is a multilayer perceptron containing N hidden layers of the same size and an output layer with one node per senone. The **bottleneck** architecture contains N hidden layers where the second-to-last hidden layer is much smaller than the others.

For the RATS experiments, we use convolutional DNNs instead of standard DNNs, because convolutional DNNs were found to give improvements in ASR and speaker verification performance relative to standard DNNs in noisy data. The first layer in a convolutional DNN consists of one or more convolutional filters followed by max-pooling. The output of each convolutional filter is a single vector whose components are obtained by taking a weighted sum of several rows of the input matrix. After the convolutional filters are applied, the resulting vectors go through a process called max pooling by which the maximum value is selected from N adjacent elements. The output vectors of the different filters after max pooling are concatenated into a long vector that is then processed by the rest of the hidden layers, which are identical to those used in the standard full or bottleneck DNN architectures. For more details on convolutional DNNs, see [23].

4) DNN Training: The DNNs are trained using alignments provided by a standard HMM-GMM ASR system where the states in the HMM are given by the senones defined by the decision tree for the target language. This system is used to align the data to the available transcriptions, resulting in a senone assignment for each frame. This assignment is used as the label during DNN training. Training is performed using a backpropagation algorithm, with small mini-batches and cross-entropy as the error metric.

5) DNN Adaptation: DNN training requires large amounts of data to achieve good performance. In this study, we use an adaptation technique for training the DNNs for low-resource languages. With this approach, a DNN trained with data from a language with sufficient resources is used to initialize the parameters of the DNN for the low-resource language. Because the last layer is language-specific (the nodes are the senones obtained with the decision tree corresponding to the language), this layer is discarded and replaced with a layer corresponding to the senones for the target language with

randomly initialized parameters. Backpropagation is then run with the data from the target language until convergence, using L2 regularization on the network's parameters to avoid overfitting to the small amount of available training data.

B. DNN-Based Features

The DNNs described above can be used to extract the two new sets of features recently proposed in the SLR literature.

1) Deep Bottleneck Features (DBFs): These features are simply the linear outputs of the bottleneck layer in a bottleneck DNN. They are frame level and relatively low dimensional. These features can be modeled like any other standard feature used for SLR (such as the MFCCs, PLPs, or SDCs) with the standard GMM-based i-vector approach. Further, as we will show, we can also model them with our proposed DNN-based i-vector approach.

2) Posterior features (DNN/post): The extraction procedure for the DNN/post features was first proposed in [7], [9]. Given a speech sample i, a DNN is used to generate the posteriors $\gamma_q(i,t)$ for every senone q and every frame t in the sample. The features are based on smooth counts provided by the DNN, computed as

$$C_q(i) = \sum_{t \in T} \gamma_q(i,t),\ q \in Q \tag{1}$$

The set of frames T can be either all frames or only speech frames as defined by a separate speech activity detection (SAD) system. The set Q can include all senones or just those senones that correspond to speech states. The discarding of non-speech senones can be interpreted as a form of smooth SAD, because hard labeling of frames as speech or non-speech is not required. In [9], we found that including all frames in the computation of C_q and discarding the non-speech senones was the optimal approach. We have also confirmed this to be the case for the LRE09 data. Hence, in this study, the DNN/post features are always computed by using this procedure.

The final features are obtained by normalizing the above counts and taking the logarithm:

$$Z_q(i) = \log\left(\frac{C_q(i)}{\sum_{s \in Q} C_s(i)}\right),\ q \in Q. \tag{2}$$

When Q is the set of speech senones only, the denominator in this equation is a smooth measure of the number of speech frames within set T. The value inside the log is a probability distribution over q. This value can be interpreted as an estimation of the posterior probability of each senone in Q for the language present in sample i.

The dimension of the resulting vector is equal to the size of the set Q, which can be much larger than the usual size of the i-vector modeled by standard backend approaches for SLR. In our first paper proposing this approach, we used probabilistic principal component analysis (PPCA) [24] to reduce the dimension of the feature vector to a size more similar to that of i-vectors. Nevertheless, in [9], we found that this step only hurts performance. Later, this finding was also confirmed for the LRE09 data. Hence, here we directly model the $Z(i)$ vector defined above.

C. GMM-Based and DNN-Based i-Vector Extraction

In the i-vector model [3], the t-th speech frame $\mathbf{x}_t^{(i)}$ from the i-th speech segment is assumed to be generated by the following GMM:

$$\mathbf{x}_t^{(i)} \sim \sum_k \gamma_{kt}^{(i)} \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{T}_k \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_k) \tag{3}$$

where the indices k are the components in the mixture (or, as we will call them, the *classes*); the $\mathbf{T}_k$ matrices describe a low-rank subspace (called the total variability subspace) by which the means of the Gaussians are adapted to a particular speech segment; $\boldsymbol{\omega}^{(i)}$ is a segment-specific, normal-distributed latent vector; $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance of the unadapted k-th Gaussian; and $\gamma_{kt}^{(i)}$ encodes the soft assignment of sample i to class k at time t. In general, we compute the assignments as the posterior of the k-th class, given the features. The i-vector used to represent the speech signal is the maximum a posteriori (MAP) point estimate of the latent vector $\boldsymbol{\omega}^{(i)}$.

Equation (3) models a process by which the feature vector for time t is generated by first choosing a class k according to the distribution $\gamma_{kt}^{(i)}$ and then generating the features according to the Gaussian distribution for that class, $\mathcal{N}(\boldsymbol{\mu}_k + \mathbf{T}_k \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_k)$. Note that the classes can be defined in any way, subject to the theoretical restriction that they have a Gaussian distribution.

Given a speech segment, the following sufficient statistics can be computed using the posterior probabilities of the classes:

$$\begin{aligned} \mathbf{N}_k^{(i)} &= \sum_t \gamma_{kt}^{(i)} \\ \mathbf{F}_k^{(i)} &= \sum_t \gamma_{kt}^{(i)} x_t^{(i)} \\ \mathbf{S}_k^{(i)} &= \sum_t \gamma_{kt}^{(i)} x_t^{(i)} {x_t^{(i)}}^T \end{aligned} \tag{4}$$

These sufficient statistics are all that is needed to train the subspace $\mathbf{T}$ and extract the i-vector $\boldsymbol{\omega}^{(i)}$. Though the training of $\mathbf{T}$ requires an iterative algorithm, i-vector extraction is done with a closed-form expression that is a function of the statistics in Equation (4) and the $\boldsymbol{\mu}_k$ in Equation (3) (the formula can be found in [3], Equation (6)). The reader is referred to [3] and [25] for more details.

Until now, we have not yet explained how the $\gamma_{kt}^{(i)}$ are obtained. In fact, the i-vector approach does not directly define how these values should be computed. In the standard GMM-based i-vector approach, we compute these values as the posterior of the k-th Gaussian, computed from the likelihood of each Gaussian by using Bayes rule to turn them into posteriors. This approach ensures that the Gaussian approximation for each class is satisfied (by definition). In [6] we proposed a new approach for computing these posteriors by redefining the classes K to be the senones, rather than the Gaussians in a GMM. The outputs of a DNN trained to estimate the posteriors for each of the senones are then used as the γs in Equation (4). We tested this approach for speaker recognition and for SLR, and showed significant gains over the GMM-based i-vector

extraction. In this approach, we make the assumption that the features for each of these senones can be accurately modeled by a single Gaussian. While this assumption is strong, our results using this approach indicate that it is a reasonable one.

Figure 1 presents a flow diagram of the proposed DNN-based i-vector hybrid framework compared to the standard GMM-based i-vector framework.

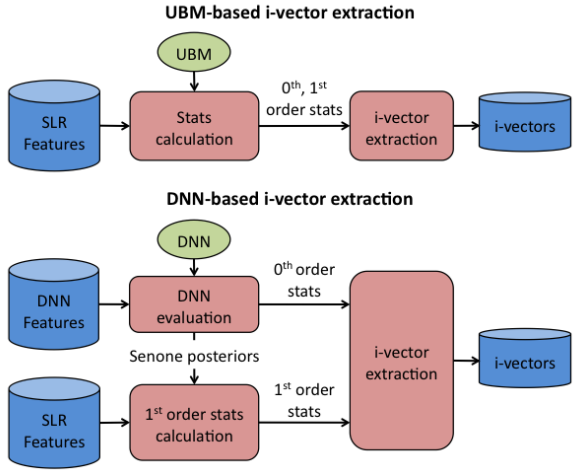

Fig. 1: Flow diagram of the GMM-based and the DNN-based i-vector approaches.

D. Backends and Calibration

In the RATS SLR task, several groups, including our own, found that a neural network outperformed other backends ([12], [26], [27]). Nevertheless, based on our own experience and that of other groups ([26], [28]), the success of the NN depends on having a large amount of training data for every target language. This is not the case for the LRE09 task. In fact, in our work with LRE09 data, we have found repeatedly, in both published and unpublished work, that the NN fails to give improvements over simpler approaches like the Gaussian backend [28]. For this reason, in our experiments, we use a NN backend for the RATS experiments and a Gaussian backend for the LRE09 experiments.

The Gaussian backend (GB) represents the state of the art in i-vector scoring for language recognition and was widely used in recent NIST Language Recognition Evaluations [29]. In the GB, a Gaussian distribution is estimated for each language, with covariance S shared across languages and language-dependent mean m_l by maximizing the likelihood on the training data. The scores are computed as the likelihood of the samples given these Gaussian models. For details on GB scoring with i-vectors see [4].

Since the Gaussian backend is used for the LRE09 task, for which training data is severely language-imbalanced, we developed a modification of the Gaussian backend approach where samples are weighted during the computation of the means and covariance of the model. Specifically, m_l and S are computed as

$$\begin{aligned} m_l &= \frac{\sum_{i|l^{(i)}=l} w^{(i)} \boldsymbol{\omega}^{(i)}}{\sum_{i|l^{(i)}=l} w^{(i)}} \\ S &= \frac{\sum_l \sum_{i|l^{(i)}=l} w^{(i)} (\boldsymbol{\omega}^{(i)} - m_l)^T (\boldsymbol{\omega}^{(i)} - m_l)}{\sum_i w^{(i)}} \end{aligned} \quad (5)$$

where $\boldsymbol{\omega}^{(i)}$ is the i-vector for sample i, $l^{(i)}$ is the language present in the sample, and $w^{(i)}$ is the weight assigned to the sample. If all weights are set to the same value, these equations coincide with those of the standard Gaussian backend. In our work, the weights are computed such that all samples from a language are weighted equally ($w^{(i)} = w^{(j)}$ if $l^{(i)} = l^{(j)}$) and the sum of their weights is identical across languages ($\sum_{i|l^{(i)}=l} w^{(i)}$ is the same for all l). Note that the sum of the weights does not affect the estimated parameters, because the language-dependent means and covariance matrix (Equations 5) only depend on the relative value of the weights. We call this backend the weighted Gaussian backend. This backend coincides with the standard Gaussian backend when all languages have the same number of samples, because the weights would be identical for all samples in that scenario.

The NN backend we use in our RATS experiments consists of a single hidden layer with 400 nodes. The activations in all layers except the last are given by hyperbolic tangent functions. In the output layer, sigmoid activations are used with one node for each target language and an additional node for the non-target language class. The NN is trained by using a backpropagation algorithm to maximize cross-entropy.

In this paper, we focus on the language detection task where, for each test sample, the system has to answer the question: "Does this sample correspond to class X?" For LRE09, we consider the closed-set condition, which was the primary condition during the evaluation. For RATS, we present open-set results, as this was the task used for evaluations. As for any detection task, optimal decisions can be made by using Bayesian decision theory if the system outputs are log-likelihood ratios (LLRs) of the null and the alternative hypothesis. LLRs can be easily obtained from the likelihood of each class given the sample [30]. Because the backends described above do not necessarily generate proper likelihoods with which to compute LLRs, a final calibration step is done to transform the scores generated by the backend into likelihoods. This transformation is done using multiclass logistic regression as described in [31]. Finally, detection LLRs are computed from the likelihoods as described in [30]. Decisions are made by thresholding these LLRs at the theoretically optimal threshold for the cost function defined in the LRE09 evaluation plan [32]. We use this cost function, commonly called Cavg (since it is an average cost across languages), multiplied by 100, to report results.

E. Architecture of the Systems under Study

Here we summarize how the techniques described above are combined to create each of the systems under comparison. In all cases, the resulting vectors are modeled with a GB or NN backend (depending on the task) and calibrated using

the multiclass logistic regression procedure described above. To avoid redundancy, these steps are not mentioned in each system description.

In all system names below, the key ***lang*** specifies the language with which the DNN was trained and the key ***feat*** specifies the feature used for i-vector extraction.

***feat* GMM/iv:** Features are processed by the standard i-vector extraction procedure based on a GMM. This approach is currently used as baseline in most SLR papers.

***feat* DNN-*lang*/iv:** Features are processed by the proposed DNN-based i-vector extraction procedure. For this system, the full DNN architecture is used, as it leads to slightly better frame-level accuracy in senone prediction than the bottleneck architecture.

DBF-*lang* GMM/iv: A bottleneck DNN trained with language *lang* is used to extract the bottleneck features, which are then modeled with the standard GMM-based i-vector procedure.

DBF-*lang* DNN-*lang*/iv: A bottleneck DNN trained with language *lang* is used to extract the bottleneck features, which are then modeled with the DNN-based i-vector procedure. The full DNN architecture is used for the modeling step.

DNN-*lang*/post: A DNN is used to estimate senone posteriors for each frame; posteriors are then processed as described in II-B2 to obtain a single vector per sample that is then modeled with a backend identical to the one used to model the i-vectors from all previous systems. Unless otherwise indicated, the full DNN architecture is used for this system.

When systems that use DNNs trained with different languages are fused at the score level, we prepend the word **parallel** to the name of the system to denote the fusion system, as this approach resembles the familiar parallel phonetic recognition approach ([15], [16]). For example, the parallel DNN/post system is a fusion of two or more DNN-*lang*/post systems.

For all RATS experiments, all DNNs are replaced by convolutional DNNs (with or without a bottleneck layer). We keep the DNN nomenclature in all cases for simplicity.

III. LRE09 Experiments

This section describes the LRE09 dataset, the system configuration used for the experiments, and results on various systems that demonstrate the value of using senone-driven DNNs for language identification under the relatively clean conditions present in this dataset. We also show the advantage of using the modified Gaussian backend proposed here, as well as the adaptation procedure for DNNs trained with languages with a small amount of training data.

A. Dataset and System Configuration

The LRE09 data includes 23 target languages. We focused on the closed-set condition, where all test samples belong to one of the 23 target languages with no out-of-set samples. The test data offers a relatively balanced number of samples per language, between 878 and 2976, for a total of 31,178 samples. Three test conditions are defined with test sample durations of approximately 3, 10 and 30 seconds.

The training data for the GMMs, i-vector extractor and backends for all systems was extracted from CallFriend, NIST LRE03, NIST LRE05, NIST LRE07, and VOA3, and contains a very unbalanced representation of the 23 target languages, ranging from 100 to 7275 samples per language. The training samples were restricted to contain at least four seconds of detected speech. All GMMs had 2048 components with diagonal covariances. The i-vectors were all of dimension 400, regardless of whether they were estimated by using the GMM/iv or the DNN/iv approaches. All systems were gender-independent in all their components.

For the LRE09 experiments, the spectral features used for the baseline system and the DNN/iv approach were shifted-delta cepstrum (SDC) features given by mel-frequency cepstral coefficients (MFCC) features of 7 dimensions appended with 7-1-3-7 shifted delta cepstra [33], resulting in a final vector of 56 dimensions. These features were pre-processed with signal-level mean and variance normalization before i-vector extraction.

DBFs were obtained by using the DNNs described below. These features were normalized the same way as the SDC features before i-vector extraction.

We used a GMM-based speech activity detection (SAD) system. Two GMMs, one for speech and one for non-speech, were trained using MFCCs of 12 dimensions plus energy, deltas, and double deltas. These GMMs were trained using data from Fisher 1 and 2, Switchboard phase 2 and 3, Switchboard cellphone phase 1 and 2, and Mixer data. Simulated noisy signals were also used for training. These signals were created by starting from a subset of clean-microphone data from the Mixer collection and by adding HVAC and babble noise as described in [34]. The annotations used for training were generated by our previous SAD system which consisted of a speech/non-speech hidden Markov model (HMM) decoder and various duration constraints. In testing, the LLR of the speech versus non-speech GMMs was found for each frame. Finally, a median filter of 21 frames was used to smooth the obtained LLRs. Frames with a smoothed LLR above 0 were declared speech.

The SLR training data described above cannot be used for DNN training because it does not include transcriptions. Hence, for this purpose, we used data from different English (eng), Mandarin (man), Spanish (spa), and colloquial Egyptian Arabic (cea) collections for which transcriptions are available. The English data came from the Fisher, CallHome, and Switchboard collections; the Mandarin data came from the CallHome, CallFriend, and the 2004 Rich Transcription evaluation collections; and the Arabic and Spanish data came from the CallHome collection. These collections contain speech collected over a telephone channel. The number of hours of data available for each language was 1300, 102, 18, and 16 for English, Mandarin, Spanish and Arabic, respectively.

The senones for which posteriors are predicted by the DNN are language-dependent and were automatically defined by using the decision tree procedure described in Section II-A1 using the data and phoneset for that language. We set the number of senones at approximately 3500 for the eng and man DNNs, and at approximately 1500 for the spa and cea DNNs,

due to the smaller amount of data available for training. The exact numbers were defined during decision tree training and were 3450 for eng, 3682 for man, 1618 for spa, and 1573 for cea. A larger number of senones for the eng DNN resulted in small degradations in SLR performance at short test durations. The DNNs contained 5 hidden layers with 1200 nodes each, except for the bottleneck DNN, which contained only 80 nodes in the bottleneck layer and 1200 in all the other layers. The size of the bottleneck layer of 80 was found to be optimal in [12], a fact that was also confirmed by our experiments. The input features to the DNNs were given by 40 log mel-filterbank coefficients with a context of 7 frames on each side of the center frame for which predictions were made.

The alignments used to train the DNNs were obtained with an HMM-GMM ASR system with 200k Gaussians that were trained to maximize the likelihood of the same data used to train the DNNs. The features used in the HMM-GMM model were 39-dimensional MFCC features, including 13 static features (including C0) and first- and second-order derivatives. The features were pre-processed with speaker-based mean and covariance normalization (MVN).

Gaussian backends were used for these experiments, as described in Section II-D. The scores generated by the backends were further calibrated through multi-class logistic regression using two-fold cross-validation on the test data.

B. System and Backend Comparison Using English-Trained DNNs

In this section we compare the five systems described in Section II-E when a DNN trained with English data is used for the systems that require it. Figure 2 shows the Cavg results for the five systems with standard (dark bars) and weighted (lighter bars) Gaussian backends. We see that the weighted backend gives a consistent improvement across all systems and durations, with the only exception being the three-second condition for the DNN/post system. Given this, all LRE09 results in the subsequent sections will use this weighted backend. The figure also shows that the DNN/iv approach with SDC features (red bars) does not outperform the baseline GMM/iv system (blue bars) using the same features. Nevertheless, the other DNN-based approaches outperform the baseline in most conditions. In particular, the approaches based on bottleneck features greatly outperform all other approaches. As for the SDC features, though, DNN/iv modeling does not outperform the GMM/iv modeling of the bottleneck features. However, the relative difference between these systems is considerably smaller.

C. Effect of the Bottleneck Layer on the DNN/post System

In the previous section we showed results using two different DNNs: one with a bottleneck layer to extract the bottleneck features, and one standard DNN with all hidden layers of the same size for the DNN/iv and the DNN/post approaches. A question arises, therefore, as to whether the bottleneck DNN can be used for these last two approaches instead of the full DNN. Is anything lost when using a bottleneck layer in the DNN? To answer this question, we ran the DNN/post approach

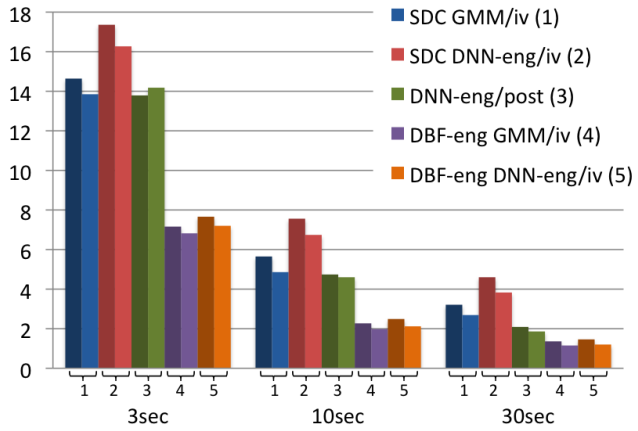

Fig. 2: Cavg×100 for LRE09 for the five systems described in Section II-E using English-trained DNNs. Dark bars correspond to the systems that use a standard Gaussian backend. Lighter bars beside each of the dark lines of the same color correspond to the same system but using the weighted Gaussian backend.

using a bottleneck and a full DNN. Results indicate a modest relative loss in performance of 3% when using the bottleneck DNN for this system instead of the full DNN at 3-second test durations, while no difference in performance is observed for 10- and 30-second test conditions.

The fact that the loss is so small implies that the bottleneck layer does not significantly hinder the DNN/post features. This might mean that the bottleneck layer is able to convey all the information needed to extract the posteriors in the output layer. If this is the case, the DBFs have an advantage over the DNN/post features in that they are lower-dimensional and can be modeled with the i-vector approach, which enables for a much more complex representation of the empirical distribution present in the samples. Though the DNN/post features model the empirical distribution by the mean over the frames, the i-vectors obtained from DBF features model the distribution through the deviations of the means of a GMM.

On the other hand, the small loss in performance observed on the DNN/post system when using a bottleneck DNN might, in fact, be due to the lack of complexity in the representation of the DNN/post features. The frame-level features from the full DNN might have more information than those from the bottleneck DNN, but our simple feature-extraction procedure that is based on the mean over the frames is unable to use of this information. This is something we plan to investigate in the near future.

D. Use of Different Languages for DNN Training

For the LRE task, we have four transcribed datasets with acoustic conditions similar to those present in the evaluation data (telephonic speech). As mentioned in Section III-A, the size of these datasets is very different, ranging from 1300 hours for the eng dataset to 16 hours for the cea dataset. In this section, we explore the use of these datasets for training the DNNs to be used for the DNN/post and DBF

GMM/iv approaches. We show results when using adaptation for DNN creation for the three languages with fewer resources. The DNNs for these languages are adapted to the English DNN by using the procedure described in Section II-A5. The regularization parameter was loosely calibrated on the DNN-cea/post system to a value of 0.01. All other systems use this same setting without recalibration.

Figure 3 shows the results for the DBF GMM/iv system when using bottleneck DNNs trained with eng, man, spa and cea. For the last three languages, results without DNN adaptation (dark bars) and with DNN adaptation (lighter bars) are shown. We can see that the performance of the systems that use unadapted DNNs directly relates to the amount of data available for training the DNN. Nevertheless, after adaptation, this correlation disappears: all systems reach approximately the performance of the one that uses the DNN trained on English, independent of the amount of data available for the target language. This result is despite the fact that the number of senones for spa and cea is approximately 1500 (even when adaptation is done) and approximately 3500 for man and eng, which confirms that SLR performance is not overly sensitive to the number of senones predicted by the DNN. Finally, we show the score-level fusion of the systems that use the eng DNN and the adapted man, cea and spa DNNs. The fusion, which we will call the **parallel DBF GMM/iv system**, leads to a relative improvement over the eng-only system of 12%, 13%, and 6% for the 3-, 10-, and 30-second conditions, respectively. These relatively small gains in fusion, and the fact that performance with the adapted DNNs is quite similar across languages, might indicate that bottleneck features are relatively independent of the language with which the DNN is trained.

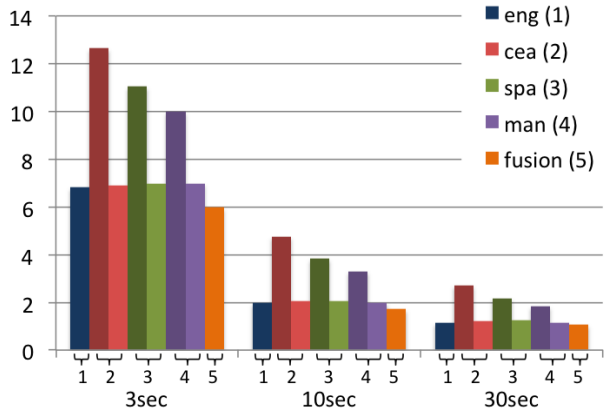

Fig. 3: Cavg×100 for LRE09 for different DBF GMM/iv systems using DNNs trained with different languages. For man, cea, and spa, two bars are shown. The darker bar corresponds to a system that uses a DNN trained directly on the data from the language. The lighter bar to the right of each darker bar corresponds to a system that uses DNNs adapted to the eng DNN. The last bar corresponds to the fusion of the systems that use the eng DNN and the adapted DNNs for man, cea, and spa.

Figure 4 shows the results for the DNN/post system when using the DNNs trained on the four languages, with and without adaptation. The conclusions for this system are identical to those for the DBF GMM/iv systems above: (1) performance of systems that use unadapted DNNs is directly related to the amount of training data for the language used for training, and (2) adaptation brings the Cavg to the same level achieved with the eng DNN. The only difference here is that the fused system, which we will call the **parallel DNN/post system**, leads to much bigger relative gains with respect to the system that uses the eng DNN, of 39%, 48%, and 34% on the 3-, 10-, and 30-second conditions, respectively. These larger gains might indicate that posterior features are inherently more language-dependent than bottleneck features. If this was the case, and we could find a way of modeling the output layer of the DNN that led to better individual performance, perhaps their fusion could outperform the fusion of DBF systems. On the other hand, perhaps the larger correlation between DBF language-dependent systems than between DNN/post language-dependent systems, which leads to the smaller gain in fusion, might be an artifact of our adaptation procedure. Because the bottleneck layer is initialized with the parameters of the eng DNN, the DBFs of the adapted DNNs are biased toward those of the eng DNN. This is not the case for the output layer, which is trained from scratch on the adapted DNNs. We plan to look into all these issues in our future research.

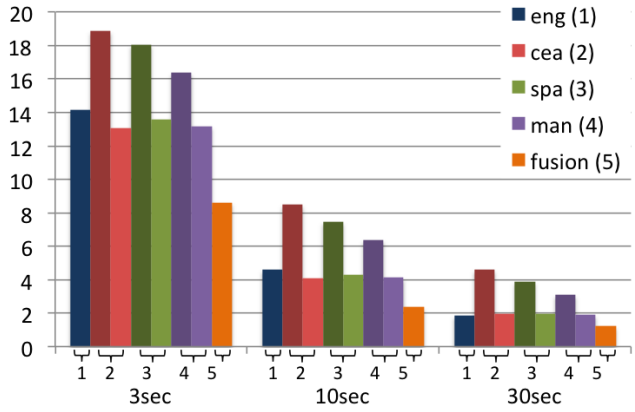

Fig. 4: Cavg×100 for LRE09 for different DNN/post systems using DNNs trained with different languages. See caption in Figure 3 for the meaning of the bars.

E. Summary of Results

Figure 5 shows three individual systems from Figure 2 (lighter bars), selecting the ones that use a weighted Gaussian backend and discarding the ones that use DNN/iv modeling, as this approach does not lead to gains over using GMM/iv modeling on this task. The figure also shows the fused systems in Figures 3 and 4 (darker bars), the parallel DBF GMM/iv and parallel DNN/post approaches. Finally, it shows the results when fusing all eight systems involved in those two fusions, or only the two systems based on eng DNNs. We can see

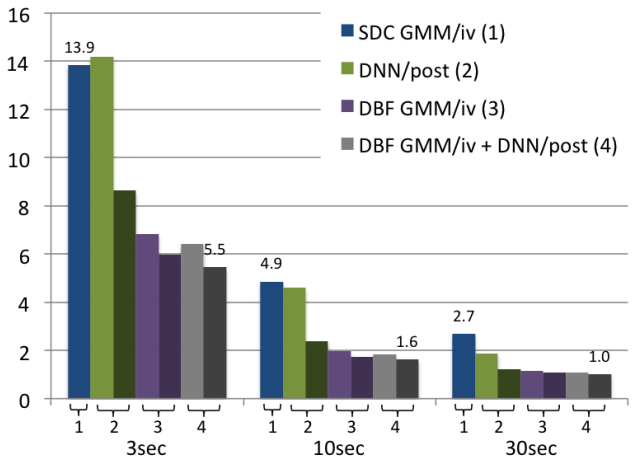

Fig. 5: Cavg×100 for LRE09 for different individual approaches and their fusion. For the DNN-based systems, two bars are shown: the lighter one to the left corresponds to a system that uses only the eng DNNs, while the darker one to the right corresponds to the parallel system (that is, a score-level fusion of systems using DNNs trained with different languages (eng, cea, spa, and man)).

System	3sec	10sec	30sec
SDC GMM/iv	13.85	4.86	2.69
DNN-eng/post	14.18	4.6	1.86
DNN-***/post	8.63	2.38	1.22
DBF-eng GMM/iv	6.82	1.98	1.15
DBF-*** GMM/iv	5.98	1.72	1.08
DBF-eng GMM/iv + DNN-eng/post	6.41	1.84	1.09
DBF-*** GMM/iv + DNN-***/post	5.45	1.62	1.02

TABLE I: Cavg×100 for LRE09 for different individual systems and fusions corresponding to those in Figure 5. The asterisks indicate that the system is a parallel system (the darker bars in Figure 5).

that, in both cases, the fusion of the two approaches leads to gains over using only the DBF approach. In particular, when all eight systems are fused from both approaches, we achieve relative gains between 6 and 9% with respect to fusing only the DBF systems. This eight-way fusion performs 60 and 67% better compared to the baseline SDC GMM/iv system. Adding the SDC GMM/iv and SDC DNN/iv systems to this fusion or to the eng-only fusion does not lead to further gains (result not shown). Results from the figure are repeated in Table I for completeness.

The results from the fused system (last line in Table I) are, by a large margin, the best results reported on this task in the literature. For example, for the three-second condition, the best reported Cavg results we could find for this task were 9.71, for a DBF-based system [10], and 10.2, for a fusion of 15 different systems [35]. At the other extreme, for the 30-second condition, the best reported Cavg result we were able to find was 1.25, for a fusion of a baseline system and a phonotactic system [36].

IV. RATS EXPERIMENTS

This section describes the RATS SLR dataset, the system configuration used for the experiments, and results on various systems that show that the main conclusions obtained on the clean LRE09 data carry over to this challenging dataset.

A. Dataset and System Configuration

The RATS SLR task consists of five target languages (Farsi, Urdu, Pashto, Levantine Arabic and Dari) and a predefined set of ten out-of-set languages ([27], [37]). Clean conversational telephone recordings were retransmitted over seven channels for the RATS program; the eighth channel, D, was excluded from the SLR task. The signal-to-noise ratio (SNR) of the retransmitted signals ranged between 30 dB to 0 dB. Four conditions were considered in which test signals were constrained to have a duration close to 3, 10, 30 and 120 seconds, respectively. The details of the task can be found in [38].

The data used for training the GMM, i-vector models, and the backends for all systems included data from the five target languages and the out-of-set languages, as well as some related languages, extracted from the RATS SLR training set. Samples from this set were selected to constitute a relatively balanced distribution of target languages, with a total of 23K segments with a mean speech duration of 80 seconds. As with the LRE experiments, all GMMs had 2048 components with diagonal covariances; the i-vectors were all of dimension 400; and all systems were gender-independent in all their components.

Given the large mismatch between the length of the training segments (80 seconds of speech on average) and the shorter test conditions, different research groups have proposed to chunk the training data to generate i-vectors using waveforms of durations closer to those of the test samples, reporting significant gains from this procedure ([9], [12], [39]). For the experiments presented here, the training dataset was chunked into segments containing approximately 8 seconds and 30 seconds of speech with 50% overlap between segments in both cases. These data, along with the full waveforms, were used for backend training. Note that the relative gains from chunking are somewhat system- and backend-dependent. As a rule of thumb, the higher the dimension of the vector that is input to the backend (400 versus the number of senones) and the more complex the backend (GB versus NN), the larger the benefit from using the chunks.

For this dataset, power-normalized cepstral coefficient (PNCC) features of 40 dimensions were used for the baseline and DNN/iv approaches [40]. These features were processed using a method proposed in [41]. The discrete cosine transform (DCT) over a window of 21 frames of PNCC features was computed. From those coefficients, a set of 80 with the highest average rank across a set of speech frames from the training data was selected. Our experiments showed that these features performed significantly better on the RATS SLR task than other features like the mel filter bank and medium-duration modulation cepstral (MDMC) features processed in this same manner. See [42] for a description of the raw MDMC and PNCC features along with more references. Because the focus

of this paper is not the features, we only show results for the best set of features.

DBFs were obtained using the DNNs described below. These features were mean and variance normalized before i-vector extraction.

As with the LRE09 experiments, we used a GMM-based SAD system. Speech and non-speech 1024-dimensional GMMs were trained using 20-dimensional PNCC features, plus deltas and double deltas, on the speaker identification data from the RATS program (which includes speech/non-speech annotations). These models were used to obtain LLRs per frame. In this case, instead of making hard decisions on whether a frame is speech or not, the LLRs were transformed via a sigmoid, with shift=-0.6 and scale of 1.0, to obtain pseudo-posteriors. These values were used when computing the accumulators for each sample to smoothly decide how much contribution each frame should have in the computation. More details on this novel "soft" SAD approach can be found in [43].

The RATS SLR data does not include transcriptions. Hence, the RATS keyword-spotting training set including 260 hours of Levantine Arabic (lev) and 400 hours of Farsi (fas) data was used to train the HMM-GMMs and the convolutional DNNs for these experiments. The data includes clean and channel-degraded waveforms. We set the number of senones for each DNN at approximately 3500. The exact number of senones after decision tree training was 3513 for fas and 3323 for lev. In [9], we showed that larger DNNs give a small performance improvement. However, we decided to use the medium size for computational reasons. Two hundred convolutional filters of size 8 were used in the convolutional layer, and the pooling size was set to three without overlap. Five hidden layers of size 1200 were used after the convolutional layer. As with the LRE experiments, the input features to the DNNs were given by 40 log mel-filterbank coefficients with a context of 7 frames on each side of the center frame for which predictions were made. No adaptation was used to train the language-specific DNNs for this task because they both have a similar amount of training data.

The alignments used to train the DNNs were obtained with an HMM-GMM ASR system with 200k Gaussians that were trained to maximize the likelihood of the same data used to train the DNNs. The features used in these models are 52-dimensional PLPs followed by heteroscedastic linear discriminant analysis (HLDA) to reduce their dimension to 39. These features were pre-processed with speaker-based MVN.

Neural network backends were used for these experiments, as described in Section II-D. The scores generated by the backends were further calibrated through multi-class logistic regression using two-fold cross-validation on the test data. The split into two folds was done based on the name of the source signal (the signal that was retransmitted to generate the distorted ones), keeping all retransmitted signals from the same source in the same split.

B. Results

Figure 6 shows results for the systems described in Section II-E. The DNN/iv system based on DBF features is not shown in the figure because its results are worse than those of the GMM/iv approach based on the same features. All DNN-based systems in this figure correspond to parallel systems where a fas-only and a lev-only system are fused at the score level. This fusion leads to gains up to 25% relative to the better of the two systems being fused. For this task, we have also tried to train multilanguage DNNs, using a merged phone space for the two languages (fas+lev). We do not show results for this approach because the parallel approach always outperforms the single system that uses the fas+lev DNN. This observation was first made in [9] for the DNN/post system. We have also confirmed this to be the case for the other DNN-based systems presented here.

Comparing with Figure 2, we can see that the main conclusion holds for both datasets: the DBF-based systems greatly outperform all other systems. We also see that modeling these features with the DNN/iv approach does not lead to consistent gains. On the other hand, we can see one important discrepancy: the DNN/iv system based on spectral features outperforms the GMM/iv system for the RATS dataset, but not for the LRE09 dataset. Note that this discrepancy is not due to the difference in features (SDCs in the case of LRE09 and PNCC in the case of RATS). The DNN/iv approach outperforms the GMM/iv approach for a variety of features on the RATS data, including MDMC, PLP and mel spectra. We believe this result can be explained by the fact that the DNN for RATS is trained on channels that perfectly match those seen in testing. On the other hand, for LRE09, the channels used for training the DNN are somewhat different from those seen in testing, which include telephone speech retransmitted over radio channels. This mismatch probably results in lower-quality posteriors from the DNN for LRE09 than for RATS, which would explain why the DNN/iv systems perform worse for LRE09 than for RATS. The DBF-based and DNN/post systems, on the other hand, do not seem to suffer from the lower-quality DNNs; they are exceptionally good for both datasets.

The figure also shows a few different fusion systems that merge different approaches. The full fusion includes all seven individual systems involved in the different approaches (one for the PNCC GMM/iv system and two, one for fas and one for lev, for each of the other systems). We can see that fusion leads to significant gains for the shortest durations. Adding systems other than the DBF GMM/iv and DNN/post systems only leads to small gains for the three-second condition.

V. Conclusions

In this paper we present and compare various approaches previously proposed in the literature for using senone-driven DNNs in SLR systems. We evaluate the systems on two tasks: the standard 2009 NIST LRE and RATS SLR. These tasks present very different challenges to the SLR systems, given their different characteristics in terms of acoustic conditions, task definition, and available training data.

The DNN/iv approach, which replaces the GMM-based posteriors during i-vector extraction with DNN-based posteriors, was first proposed for speaker recognition and leads to impressive gains on telephone speech. When applied to SLR tasks,

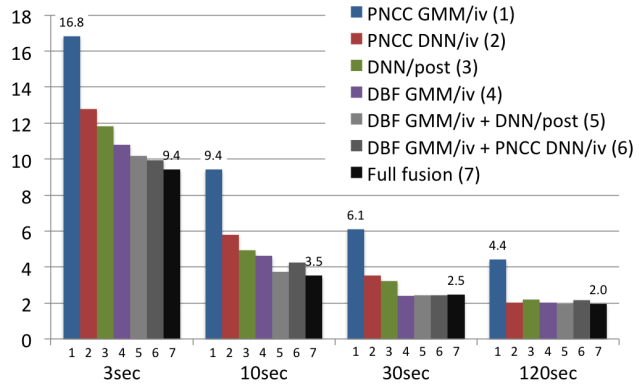

Fig. 6: Cavg×100 for RATS SLR for different individual approaches and their fusion. All DNN-based systems are parallel systems (that is, a score-level fusion of two systems, one based on fas, and one based on lev DNNs).

the approach outperforms the baseline GMM/iv approach for the RATS task but not for the LRE09 task. We believe this result is due to differences in channel characteristics between the data used for training the DNNs for LRE09, which is restricted to telephone speech, and the evaluation data, which includes telephone speech retransmitted over radio channels. On the other hand, the data used to train the RATS DNNs is well matched to the evaluation data, including the same transmission channels. As we have observed for the speaker recognition task, this approach appears to be quite sensitive to data mismatch.

The other two approaches tested here use a senone-driven DNN to extract features, which are then modeled with standard techniques. The first approach extracts the features from a bottleneck layer in the DNN; the second one uses the output layer. We show that the approach where bottleneck features are modeled with the standard GMM/iv technique outperforms all other approaches on both the LRE09 and the RATS task, followed closely by the approach based on the posterior features extracted from the output layer of the DNN. Both approaches greatly outperform the baseline GMM/iv system based on standard spectral features and also outperform the DNN/iv approach. We believe one key to the success of these two systems is that the features extracted this way are robust to channel and speaker variation, given that the DNNs used to extract them are trained to predict the senone posteriors for a large set of speakers and channels.

For the LRE09 task, where the training data is quite unbalanced across languages, we propose a weighted version of the Gaussian backend that leads to consistent gains across all systems and durations, with negligible additional computational cost. We also propose using an adaptation procedure for training DNNs for languages with few resources. This procedure leads to relative gains of up to 57% on the bottleneck and posterior feature approaches. Fusion of systems based on the adapted DNNs from four different languages leads to significant gains on both approaches, with much larger gains on the posterior-based approach.

Overall, the fusion of bottleneck and posterior feature-based approaches with DNNs trained with different languages results in systems that are between 40 and 70% better than the baseline GMM/iv systems for both tasks over all test durations.

In the future, we plan to explore alternative feature creation and modeling techniques for the output layer of the DNN. We will also evaluate recent deep learning techniques for senone posterior estimation that have led to improvements in ASR, such as deep convex networks and long short-term memory recurrent neural networks.

REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[2] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. ASLP*, vol. 20, pp. 30–42, 2012.

[3] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[4] D. Martínez-González, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," in *Proc. Interspeech*, Florence, Italy, Aug. 2011.

[5] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction.," in *Proc. Interspeech*, Florence, Italy, Aug. 2011.

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. ICASSP*, Florence, Italy, May 2014.

[7] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. Odyssey-14*, Joensuu, Finland, June 2014.

[8] P Kenny, V Gupta, T Stafylakis, P Ouellet, and J Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey-14*, Joensuu, Finland, June 2014.

[9] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Spoken language recognition based on senone posteriors," in *Proc. Interspeech*, Singapore, Sept. 2014.

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

[11] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PLOS One*, July 2014.

[12] P. Matejka, L. Zhang, T. Ng, S. H. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," in *Proc. Odyssey-14*, Joensuu, Finland, June 2014.

[13] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martínez-González, J. Gonzalez-Rodriguez, and PJ Moreno, "Automatic language identification using deep neural networks," in *Proc. ICASSP*, Florence, Italy, May 2014.

[14] M. Diez, A. Varona, M. Penagarikano, L. J. Rodriguez-Fuentes, and G. Bordel, "On the use of log-likelihood ratios as features in spoken language recognition," in *IEEE Workshop on Spoken Language Technology (SLT 2012)*, Miami, Florida, USA, 2012.

[15] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Interspeech-2005*, 2005.

[16] W. Shen, W. Campbell, T. Gleason, D. Reynolds, and E. Singer, "Experiments with lattice-based PPRLM language identification," in *Proc. Odyssey-06*, Puerto Rico, USA, June 2006.

[17] A. Stolcke, M. Akbacak, L. Ferrer, S. Kajarekar, C. Richey, N. Scheffer, and E. Shriberg, "Improving language recognition with multilingual phone recognition and speaker adaptation transforms," in *Proc. Odyssey-10*, Brno, Czech Republic, June 2010.

[18] L. F. D'Haro, O. Glembek, O. Plchot, P. Matejka, M. Soufifar, R. Cordoba, and J. Cernocky, "Phonotactic language recognition using i-vectors and phoneme posteriogram counts," in *Proc. Interspeech*, Portland, USA, Sept. 2012.

[19] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *HLT '94 Proceedings of the workshop on Human Language Technology*, 1994.
[20] L. Deng and D. Yu, "Deep convex network: A scalable architecture for speech pattern classification," in *Proc. Interspeech*, Florence, Italy, Aug. 2011.
[21] P. Huang, L. Deng, M. Hasegawa-Johnson, and X. He, "Random features for kernel deep convex network," in *Proc. ICASSP*, Vancouver, May 2013.
[22] A. Mohamed A. Graves, N. Jaitly, "Hybrid speech recognition with deep bidirectional LSTM," in *Proceedings IEEE Workshop on Speech Recognition and Understanding*, Olomouc, Czech Republic, Dec. 2013.
[23] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," *MIT Press*, pp. 255–258, 1995.
[24] N. Scheffer, Y. Lei, and L. Ferrer, "Factor analysis back ends for MLLR transforms in speaker recognition," in *Proc. Interspeech*, Florence, Italy, Aug. 2011.
[25] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 16, pp. 980–988, July 2008.
[26] P. Matějka, O. Plchot, M. Soufifar, O. Glembek, L. F. D'haro Enríquez, K. Veselỳ, F. Grézl, J. Ma, S. Matsoukas, and N. Dehak, "Patrol team language identification system for DARPA RATS P1 evaluation," in *Proc. Interspeech*, Portland, USA, Sept. 2012.
[27] A. Lawson, M. McLaren, Y. Lei, V. Mitra, N. Scheffer, L. Ferrer, and M. Graciarena, "Improving language identification robustness to highly channel-degraded speech through multiple system fusion," in *Proc. Interspeech*, Lyon, France, Aug. 2013.
[28] M. McLaren, A. Lawson, Y. Lei, and N. Scheffer, "Adaptive gaussian backend for robust language identification," in *Proc. Interspeech*, Lyon, France, Aug. 2013.
[29] M. Penagarikano, A. Varona, M. Diez, L. J. Rodriguez-Fuentes, and G. Bordel, "Study of different backends in a state-of-the-art language recognition system," in *Proc. Interspeech*, Portland, USA, Sept. 2012.
[30] N. Brummer and D. A. van Leeuwen, "On calibration of language recognition scores," in *Proc. Odyssey-06*, Puerto Rico, USA, June 2006.
[31] D. A. Van Leeuwen and N. Brummer, "Channel-dependent GMM and multi-class logistic regression models for language recognition," in *Proc. Odyssey-06*, Puerto Rico, USA, June 2006.
[32] "NIST LRE09 evaluation plan," http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf.
[33] B. Bielefeld, "Language identification using shifted delta cepstrum," in *Fourteenth Annual Speech Research Symposium*, 1994.
[34] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer, "Promoting robustness for speaker modeling in the community: The PRISM evaluation set," in *Proceedings of SRE11 Analysis Workshop*, Atlanta, Dec. 2011.
[35] Z. Jancik, O. Plchot, N. Brümmer, L. Burget, O. Glembek, V. Hubeika, M. Karafiát, P. Matejka, T. Mikolov, A. Strasheim, et al., "Data selection and calibration issues in automatic language recognition-investigation with BUT-AGNITIO NIST LRE 2009 system," in *Proc. Odyssey-10*, Brno, Czech Republic, June 2010.
[36] L. F. D'haro Enríquez, O. Glembek, O. Plchot, P. Matějka, M. Soufifar, R. Córdoba Herralde, and J. Černockỳ, "Phonotactic language recognition using i-vectors and phoneme posteriogram counts," in *Proc. Interspeech*, Portland, USA, Sept. 2012.
[37] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012.
[38] "DARPA RATS program," http://www.darpa.mil/Our_Work/I2O/Programs/Robust_Automatic_Transcription_of_Speech_(RATS).aspx.
[39] J. Z. Ma, B. Zhang, S. Matsoukas, S. H. R. Mallidi, F. Li, and H. Hermansky, "Improvements in language identification on the rats noisy speech corpus.," in *Proc. Interspeech*, Lyon, France, Aug. 2013.
[40] C. Kim and R.M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*, Kyoto, Mar. 2012.
[41] M. McLaren and Y. Lei, "Improved speaker recognition using DCT coefficients as features," in *Proc. ICASSP*, Brisbane, Australia, May 2015.
[42] M. McLaren, N. Scheffer, M. Graciarena, L. Ferrer, and Y. Lei, "Improving speaker identification robustness to highly channel-degraded speech through multiple system fusion," in *Proc. ICASSP*, Vancouver, May 2013.
[43] M. McLaren, M. Graciarena, and Y. Lei, "Softsad: Integrated frame-based speech confidence for speaker recognition," in *Proc. ICASSP*, Brisbane, Australia, May 2015.

Luciana Ferrer , Ph.D., is a researcher at the National Scientific and Technical Research Council (CONICET), Argentina, working at the Computer Science Department in the Exact and Natural Sciences School, University of Buenos Aires. Prior to returning to Argentina, Luciana worked at SRI International's Speech Technology and Research (STAR) Laboratory. Luciana's current work is done in close collaboration with this group. Her current research interests include speaker and language identification, speech activity detection, and pronunciation scoring for second language learning. Luciana received the B.S. degree from the University of Buenos Aires, Argentina, in 2001, and her Ph.D. degree from Stanford University in 2009.

Yun Lei , Ph.D., received the B.S. degree in electrical engineering from Nanjing University, Jiangsu, China, in 2003, the M.S. degree in electrical engineering from Institute of Acoustics, Chinese Academy of Science (CAS), Beijing, China, in 2006, and the Ph.D. degree in electrical engineering from Universitcy of Texas at Dallas. He is currently a research scientist in Facebook, Inc. Before that, he was a research engineering in SRI International from 2010 to 2015.

Mitchell McLaren , Ph.D., is a research engineer in SRI International's Speech Technology and Research (STAR) Laboratory. His research interests include speaker and language identification, as well as other biometrics such as face recognition. Prior to joining SRI, Mitchell was a postdoctoral researcher and the University of Nijmegen, The Netherlands, where he focused on speaker and face identification on the Bayesian Biometrics for Forensics (BBfor2) project, funded by Marie Curie Action. His Ph.D. in speaker identification is from the Queensland University of Technology (QUT), Brisbane, Australia.

Nicolas Scheffer , Ph.D., is currently Research Scientist at Facebook, Inc. working on automatic speech recognition, speaker and language identification. Previously, Nicolas was Sr. Research engineer at SRI Internationals Speech Technology and Research (STAR) Laboratory for 7 years. He was responsible for the technical direction for speaker and language recognition technologies. Nicolas obtained his computer science PhD at Avignon University, France. He obtained a MSc / MEng in EE, Control systems and applied computer science from Ecole des Mines / Ecole Centrale, France.