MULTILINGUAL SPEECH RECOGNITION WITH A SINGLE END-TO-END MODEL

*Shubham Toshniwal**

Toyota Technological Institute at Chicago
shtoshni@ttic.edu

Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, Kanishka Rao

Google Inc., U.S.A
{tsainath, ronw, boboli, pedro, weinstein, kanishkarao}@google.com

ABSTRACT

Training a conventional automatic speech recognition (ASR) system to support multiple languages is challenging because the sub-word unit, lexicon and word inventories are typically language specific. In contrast, sequence-to-sequence models are well suited for multilingual ASR because they encapsulate an acoustic, pronunciation and language model jointly in a single network. In this work we present a single sequence-to-sequence ASR model trained on 9 different Indian languages, which have very little overlap in their scripts. Specifically, we take a union of language-specific grapheme sets and train a grapheme-based sequence-to-sequence model jointly on data from all languages. We find that this model, which is not explicitly given any information about language identity, improves recognition performance by 21% relative compared to analogous sequence-to-sequence models trained on each language individually. By modifying the model to accept a language identifier as an additional input feature, we further improve performance by an additional 7% relative and eliminate confusion between different languages.

Index Terms— ASR, speech recognition, multilingual, encoder-decoder, seq2seq, Indian

1. INTRODUCTION

Speech recognition has made remarkable progress in the past few years with services such as Google Voice Search supporting about 120 languages.[1] Further expanding its coverage of the world's $\approx$7,000 languages is of great interest to both academia and industry. However, in many cases the resources available to train large vocabulary continuous speech recognizers are severely limited [1]. These challenges have meant that there has been a perennial interest in multilingual and cross-lingual models which allow for knowledge transfer across languages, and thus relieve burdensome data requirements [2–12].

Most of the previous work on multilingual speech recognition has been limited to making the acoustic model (AM) multilingual [3–6, 9–11, 13, 14]. Some of the multilingual AMs require a common phone set [3, 4, 13] while others share some of the acoustic model parameters [9–11, 15]. A hat swap structure is proposed in [9–11], where the lower layers of a deep neural network (DNN) are shared across languages and the output layer is language-specific. Alternatively, multilingual bottleneck features from a DNN feature extractor can be used for either a Gaussian Mixture Model or DNN-based systems [16]. These multilingual AMs still require language-specific pronunciation models (PMs) and language models (LMs) which means that often such models must know the speech language identity during inference [9–11]. Moreover, the AMs, PMs and LMs are usually optimized independently, in which case errors from one component propagate to subsequent components in a way that was not seen during training.

Sequence-to-sequence models fold the AM, PM and LM into a single network, making them attractive to explore for multilingual speech recognition. Building a multilingual sequence-to-sequence model requires taking the union over all the language-specific grapheme sets and training the model jointly on data from all the languages. In addition to their simplicity, the end-to-end nature of such models means that all of the model parameters contribute to handling the variations between different languages. Our attention-based sequence-to-sequence model is based on the Listen, Attend and Spell (LAS) model [17, 18], the details of which are explained in the next section. Our work is most similar to that of [12] which similarly proposes an end-to-end trained multilingual recognizer to directly predict grapheme sequences in 10 distantly related languages. They utilize a hybrid attention/connectionist temporal classification model integrated with an independently trained grapheme LM. In this paper we use a simpler sequence-to-sequence model without an explicit LM, and study a corpus of 9 more closely related Indian languages.

We show that a LAS model jointly trained across data from 9 Indian languages without any explicit language specification consistently outperforms monolingual LAS models trained independently on each language. Even without explicit language specification, the model is rarely confused between languages. We also experiment with certain language-dependent variants of the model. In particular, we obtain the largest improvement by conditioning the encoder on the speech language identity. We also run several experiments on synthesized data to gain insights into the behavior of these models. We find that the multilingual model is unable to code-switch between languages, indicating that the language model is dominating the acoustic model. Finally, we find that the language-conditioned model is able to transliterate Urdu speech into Hindi text, suggesting that the model has learned an internal representation which disentangles the underlying acoustic-phonetic content from the language.

2. MODEL

In this section we describe the Listen, Attend and Spell (LAS) attention-based sequence-to-sequence ASR model proposed by Chan et al [17], as well as our proposed modifications to support recognition in multiple languages.

2.1. LAS Model

The sequence-to-sequence model consists of three modules: an *encoder*, *decoder* and *attention network* which are trained jointly to

*Work done at Google NYC.

[1] https://www.blog.google/products/search/type-less-talk-more/

predict a sequence of graphemes from a sequence of acoustic feature frames.

We use 80-dimensional log-mel acoustic features computed every 10ms over a 25ms window. Following [19] we stack 8 consecutive frames and stride the stacked frames by a factor of 3. This downsampling enables us to use a simpler encoder architecture than [17].

The encoder is comprised of a stacked bidirectional recurrent neural network (RNN) [20, 21] that reads acoustic features $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$ and outputs a sequence of high-level features (hidden states) $\boldsymbol{h} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K)$. The encoder is similar to the acoustic model in an ASR system.

The decoder is a stacked unidirectional RNN that computes the probability of a sequence of characters $\boldsymbol{y}$ as follows:

$$P(\boldsymbol{y}|\boldsymbol{x}) = P(\boldsymbol{y}|\boldsymbol{h}) = \prod_{t=1}^{T} P(y_t|\boldsymbol{h}, \boldsymbol{y_{<t}}).$$

The conditional dependence on the encoder state vectors $\boldsymbol{h}$ is represented by context vector $\boldsymbol{c}_t$, which is a function of the current decoder hidden state and the encoder state sequence:

$$u_{it} = \boldsymbol{v}^\top \tanh(\boldsymbol{W_h} \boldsymbol{h}_i + \boldsymbol{W_d} \boldsymbol{d}_t + \boldsymbol{b}_\text{a})$$

$$\boldsymbol{\alpha}_t = \text{softmax}(\boldsymbol{u}_t) \qquad \boldsymbol{c}_t = \sum_{i=1}^{K} \alpha_{it} \boldsymbol{h}_i$$

where the vectors $\boldsymbol{v}, \boldsymbol{b}_\text{a}$ and the matrices $\boldsymbol{W_h}, \boldsymbol{W_d}$ are learnable parameters; $\boldsymbol{d}_t$ is the hidden state of the decoder at time step t.

The hidden state of the decoder, $\boldsymbol{d}_t$, which captures the previous character context $\boldsymbol{y_{<t}}$, is given by:

$$\boldsymbol{d}_t = \text{RNN}(\tilde{\boldsymbol{y}}_{t-1}, \boldsymbol{d}_{t-1}, \boldsymbol{c}_{t-1})$$

where $\boldsymbol{d}_{t-1}$ is the previous hidden state of the decoder, and $\tilde{\boldsymbol{y}}_{t-1}$ is a character embedding vector for y_{t-1}, as is typical practice in RNN-based language models. The decoder is analogous to the language model component of a pipeline system for ASR. The posterior distribution of the output at time step t is given by:

$$P(y_t|\boldsymbol{h}, \boldsymbol{y_{<t}}) = \text{softmax}(\boldsymbol{W}_\text{s}[\boldsymbol{c}_t; \boldsymbol{d}_t] + \boldsymbol{b}_\text{s}),$$

where $\boldsymbol{W}_\text{s}$ and $\boldsymbol{b}_\text{s}$ are again learnable parameters. The model is trained to optimize the discriminative loss:

$$L_\text{LAS} = -\log(P(\boldsymbol{y}|\boldsymbol{x}))$$

2.2. Multilingual Models

In the multilingual scenario, we are given n languages $\{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$, each with independent character sets $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_n\}$ and training sets $\{(\mathcal{X}_1, \mathcal{Y}_1), \ldots, (\mathcal{X}_n, \mathcal{Y}_n)\}$. The combined training dataset is thus given by the union of the datasets for each language:

$$(\mathcal{X}, \mathcal{Y}) = \cup_{i=1}^{n} (\mathcal{X}_i, \mathcal{Y}_i)$$

and the character set for the combined dataset is similarly given by:

$$\mathcal{C} = \cup_{i=1}^{n} \mathcal{C}_i$$

2.2.1. Joint

We begin by training a joint model, consisting of the LAS model described in the previous section trained directly on the combined multilingual dataset. This model is not given any explicit indication that the training dataset is composed of different languages. However, as we will show later, this model is still able to recognize speech in multiple languages despite the lack of runtime language-specification.

2.2.2. Multitask

We also experiment with a variant of the joint model which has the same architecture but is trained in a multitask learning (MTL) configuration [22] to jointly recognize speech and simultaneously predict its language. The language ID annotation is thus utilized during training, but is not passed as an input during inference. In order to predict the language ID, we average the encoder output h across all time frames to compute an utterance-level feature. This averaged feature is then passed to a softmax layer to predict the likelihood of the speech belonging to each language:

$$p(\mathcal{L}|\boldsymbol{x}) = \text{softmax}(\boldsymbol{W}_\text{lang} \tfrac{1}{K} \Sigma_i \boldsymbol{h}_i + \boldsymbol{b}_\text{lang})$$

The language identification loss is given by:

$$L_\text{LID} = -\log(p(\mathcal{L} = \mathcal{L}_j|\boldsymbol{x})$$

where the j-th language, $\mathcal{L}_j$, is the ground truth language. The two losses are combined using an empirically determined weight λ to obtain the final training loss:

$$L_\text{MTL} = \tfrac{1}{1+\lambda} L_\text{LAS} + \tfrac{\lambda}{1+\lambda} L_\text{LID}$$

2.2.3. Conditioned

Finally, we consider a set of conditional models which utilize the language ID during inference. Intuitively, we expect that a model which is explicitly conditioned on the speech language will have an easier time allocating its capacity appropriately across languages, speeding up training and improving recognition performance.

Specifically, we learn a fixed-dimensional language embedding for each language to condition different components of the basic joint model on language ID. This conditioning is achieved by feeding in the language embedding as an input to the first layer of encoder, decoder or both giving rise to (a) *Encoder-conditioned*, (b) *Decoder-conditioned*, and (c) *Encoder+Decoder-conditioned* variants. In contrast to the MTL model, the language ID is not used as part of the training cost.

3. EXPERIMENTAL SETUP

Table 1: Multilingual dataset statistics.

Language	# training utts.	# test utts.
Bengali	364617	14679
Gujarati	243390	14935
Hindi	213753	14718
Kannada	192523	14765
Malayalam	285051	14095
Marathi	227092	13898
Tamil	164088	9850
Telugu	232861	14130
Urdu	196554	14486
Total	2119929	125556

3.1. Data

We conduct our experiments on data from nine Indian languages shown in Table 1, which corresponds to a total of about 1500 hours of training data and 90 hours of test data. The nine languages have little

overlap in their character sets, with the exception of Hindi and Marathi which both use the Devanagari script. The small overlap means that the output vocabulary for our multilingual models, which is union over character sets, is also quite large, containing 964 characters. Separate validation sets of around 10k utterances per language are used for hyperparameter tuning. All the utterances are dictated queries collected using desktop and mobile devices.

3.2. Model and Training Details

As a baseline, we train nine monolingual models independently on data for each language. We tune the hyperparameters on Marathi and reuse the optimal configuration to train models for the remaining languages. The best configuration for Marathi uses a 4 layer encoder comprised of 350 bidirectional long short-term memory (biLSTM) cells (i.e. 350 cells in forward layer and 350 cells in backward layer), and a 2 layer decoder containing 768 LSTM cells in each layer. For regularization, we apply a small L2 weight penalty of 1e-6 and add Gaussian weight noise [23] with standard deviation of 0.01 to all parameters after 20k training steps. All the monolingual models converge within 200-300k gradient steps.

Since the multilingual training corpus is much larger, we were able to train a joint larger multilingual model without overfitting. As with the training set, the validation set is also a union of the language-specific validation sets. The best configuration uses a 5 layer encoder comprised of 700 biLSTM cells, and a 2 layer decoder containing 1024 LSTM cells in each layer. For the multitask model, we find $\lambda = 0.01$ among $\{0.1, 0.01\}$ to work the best. We restricted ourselves to these values because for a very large λ, the language ID prediction task would dominate the primary task of ASR, while for a very small λ the additional task would have no effect on the training loss. For all conditional models, we use a 5-dimensional language embedding. For regularization we add Gaussian weight noise with standard deviation of 0.0075 after 25k training steps. All multilingual models are trained for approximately 2 million steps.

All models are implemented in TensorFlow [24] and trained using asynchronous stochastic gradient descent [25] using 16 workers. The initial learning rate is set to 1e-3 for the monolingual models and 1e-4 for the multilingual models with learning rate decay in all the models.

4. RESULTS

Table 2: WER(%) of language-specific, joint, and joint+MTL LAS models.

Language	Language-specific	Joint	Joint + MTL
Bengali	19.1	16.8	**16.5**
Gujarati	26.0	**18.0**	18.2
Hindi	16.5	**14.4**	**14.4**
Kannada	35.4	**34.5**	34.6
Malayalam	44.0	36.9	**36.7**
Marathi	28.8	27.6	**27.2**
Tamil	13.3	10.7	**10.6**
Telugu	37.4	**22.5**	22.7
Urdu	29.5	26.8	**26.7**
Weighted Avg.	29.05	**22.93**	**22.91**

We first compare the language-specific LAS models with the joint LAS model trained on all languages. As shown in Table 2, the joint LAS model outperforms the language-specific models for all the languages. In fact, the joint model decreases weighted average WERs across all the 9 languages, weighted by number of words, by more than 21% relative to the monolingual models. This result is quite interesting not only because the joint model is a single model that is being compared to 9 different monolingual models, but unlike the monolingual models the joint model it not language-aware at runtime. Finally, the large performance gain of the joint model is also attributable to the fact that the Indian languages are very similar in the phonetic space [26], despite using different grapheme sets.

Second, we compare the joint LAS model with the multitask trained variant. As shown in the right two columns of Table 2, the MTL model shows limited improvements over the joint model. This might be due to the following reasons: (a) Static choice of λ. Since the language ID prediction task is easier than ASR, a dynamic λ which is high initially and decays over time might be better suited, and (b) The language ID prediction mechanism of averaging over encoder outputs might not be ideal. A learned weighting of the encoder outputs, similar to the attention module, might be better suited for the task.

Table 3: WER(%) of joint LAS model and the joint language-conditioned models, namely decoder-conditioned (Dec), encoder-conditioned (Enc), and encoder+decoder-conditioned (Enc + Dec).

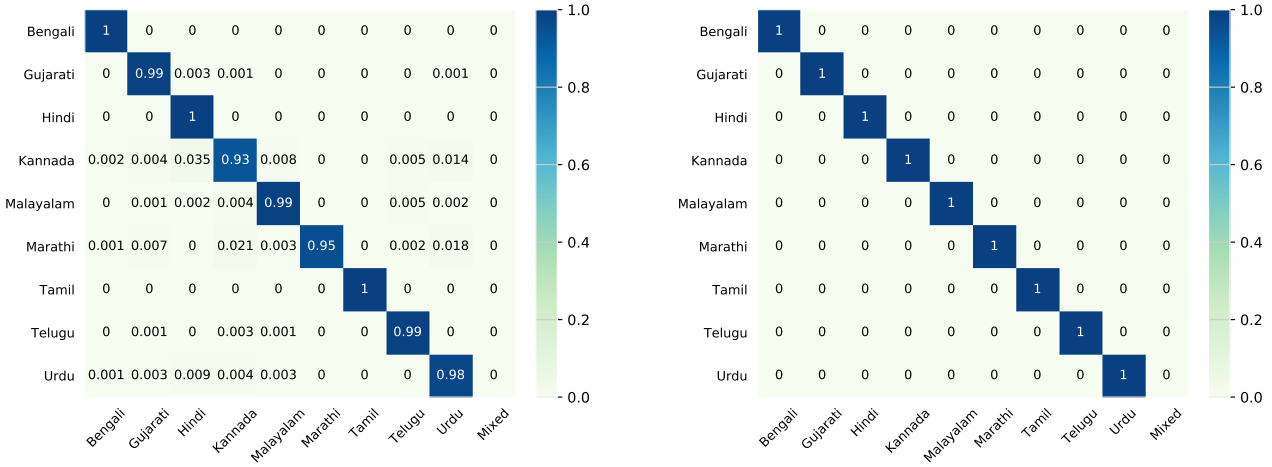
Language	Joint	Dec	Enc	Enc + Dec
Bengali	16.8	16.9	**16.5**	**16.5**
Gujarati	18.0	17.7	**17.2**	17.3
Hindi	**14.4**	14.6	14.5	**14.4**
Kannada	34.5	30.1	29.4	**29.2**
Malayalam	36.9	35.5	34.8	**34.3**
Marathi	27.6	24.0	**22.8**	23.1
Tamil	10.7	10.4	**10.3**	10.4
Telugu	22.5	22.5	21.9	**21.5**
Urdu	26.8	25.7	**24.2**	24.5
Weighted Avg.	22.93	22.03	21.37	**21.32**

Third, Table 3 shows that all the joint models conditioned on the language ID outperform the joint model. The encoder-conditioned model (Enc) is better than the decoder-conditioned model (Dec) indicating that some form of acoustic model adaptation towards different languages and accents occurs when the encoder is conditioned. In addition, conditioning both the encoder and decoder (Enc + Dec) does not improve much over conditioning just the encoder, suggesting that feeding the encoder with language ID information is sufficient, as the encoder outputs are then fed to the decoder anyways via the attention mechanism.

Comparing model performances across languages we see that all the models perform worst on Malayalam and Kannada. We hypothesize that this has to do with the *agglutinative* nature of these languages which makes the average word longer in these languages compared to languages like Hindi or Gujarati. For example, an average training set word in Malayalam has 9 characters compared to 5 in Hindi. In fact, we found that in contrast to the WER, the character error rate (CER) for Hindi and Malayalam were quite close.

5. ANALYSIS

In this section we investigate the behavior and capacity of the proposed system in more detail, by asking the questions detailed below.



(a) Joint

(b) Encoder-conditioned

Fig. 1: Confusion matrices for joint and encoder-conditioned models, truncated to precision of 10^{-3}. The joint model is rarely confused between languages, while conditioning removes those rare cases almost completely.

How often does the model confuse between languages? The ability of the proposed model to recognize multiple languages comes with the potential side effect of confusing the languages. The lack of script overlap between Indian languages, with the exceptions of Hindi and Marathi, means that the surface analysis of the script used in the model output is a good proxy to tell if the model is confused between languages or not. We carry out this analysis at the word level and check if the output words use graphemes from a single language or a mixture. We test the word first on the ground truth language, and in case of failure, test it on other languages. If the word cannot be expressed using the character set of any single language, we classify it as *mixed*. The result for both the joint and the encoder-conditioned model is summarized in Figure 1. While both the models are rarely confused between languages, the result for the joint model is interesting given its lack of explicit language awareness, showing that the LAS model is implicitly learning to predict language ID. It is also interesting to observe that by conditioning the joint model on the language ID, there is no confusion between languages.

Can the joint model perform code-switching? The joint model in theory has the capacity to switch between languages. In fact, it can code-switch between English and the 9 Indian languages due to the presence of English words in the training data[2]. We were interested in testing if the model could also code-switch between a pair of Indian languages which was not seen during training. For this purpose, we created an artificial dataset by selecting about 1,000 Tamil utterances and appending them with the same number of Hindi utterances with a 50ms break in between. To our disappointment, the model is not able to code-switch at all. It picks one of the two scripts and sticks with it. Manual inspection shows that: (a) when the model chooses Hindi, it only transcribes the Hindi part of the utterance (b) similarly when the model chooses Tamil it only transcribes the Tamil part, but on rare occasions it also transliterates the Hindi part. This suggests that the language model is dominating the acoustic model and points to overfitting, which is a known issue with attention-based sequence-to-sequence models [27].

[2]1-6% of the total words in the training set are English words in all the 9 languages.

What does the conditioned model output for mismatched language ID? The interesting question here is does the model obey acoustics or is it faithful to the language ID. To answer this, we created an artificial dataset of about 1,000 Urdu utterances labeled with the Hindi language ID and transcribed it with the encoder-conditioned model. As it turns out, the model is extremely faithful to the language ID and sticks to Hindi's character set. Manual inspection of the outputs reveals that the model transliterates Urdu utterances in Hindi, suggesting that the model has learned an internal representation which disentangles the underlying acoustic-phonetic content from the language identity.

6. CONCLUSION

We present a sequence-to-sequence model for multilingual speech recognition which is able to recognize speech without any explicit language specification. We also propose simple variants of the model conditioned on language identity. The proposed model and its variants substantially outperform baseline monolingual sequence-to-sequence models for all languages, and rarely chooses the incorrect grapheme set in its output. The model, however, cannot handle code-switching, suggesting that the language model is dominating the acoustic model. In future work, we would like to integrate the conditional variants of the model with separate language-specific language models to further improve recognition accuracy. We would also like to compare the proposed models against traditional models on live traffic data. The exploration of reasons for lack of code-switching in joint model can also lead to interesting insights regarding sequence-to-sequence models.

7. ACKNOWLEDGEMENTS

We would like to thank Rohit Prabhavalkar, Yonghui Wu, Vijay Peddinti, Zhifeng Chen and Patrick Nguyen for helpful comments. We are also thankful to the anonymous reviewers for their helpful comments.

8. REFERENCES

[1] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz, "Automatic Speech Recognition for Under-Resourced Languages: A Survey," *Speech Communication*, vol. 56, 2014.

[2] Fuliang Weng, Harry Bratt, Leonardo Neumeyer, and Andreas Stolcke, "A Study of Multilingual Speech Recognition," in *Proc. Eurospeech*, 1997.

[3] T. Schultz and A. Waibel, "Fast Bootstrapping of LVCSR Systems with Multilingual Phoneme Sets," in *Proc. Eurospeech*, 1997.

[4] Tanja Schultz and Alex Waibel, "Language-Independent and Language-Adaptive Acoustic Modeling for Speech Recognition," *Speech Communication*, vol. 35, no. 1-2, 2001.

[5] Thomas Niesler, "Language-Dependent State Clustering for Multilingual Acoustic Modelling," *Speech Communication*, vol. 49, no. 6, 2007.

[6] Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee, "A Study on Multilingual Acoustic Modeling for Large Vocabulary ASR," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.

[7] Lukas Burget, Petr Schwarz, Mohit Agarwal, Pinar Akyazi, Kai Feng, Arnab Ghoshal, Ondrej Glembek, Nagendra Goel, Martin Karafiat, Daniel Povey, Ariya Rastrow, Richard C. Rose, and Samuel Thomas, "Multilingual Acoustic Modeling for Speech Recognition Based on Subspace Gaussian Mixture Models," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010.

[8] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Multilingual MLP Features for Low Resource LVCSR Systems," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[9] Georg Heigold, Vincent Vanhoucke, Andrew Senior, Patrick Nguyen, Marc'aurelio Ranzato, Matthieu Devin, and Jeff Dean, "Multilingual Acoustic Models Using Distributed Deep Neural Networks," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[10] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual Training of Deep Neural Networks," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[11] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-Language Knowledge Transfer Using Multilingual Deep Neural Network with Shared Hidden Layers," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[12] Shinji Watanabe, Takaaki Hori, and John Hershey, "Language Independent End-to-End Architecture for Joint Language and Speech Recognition," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017.

[13] N. T. Vu, D. Imseng, D. Povey, P. Motlicek, T. Schultz, and H. Bourlard, "Multilingual Deep Neural Network Based Acoustic Modeling for Rapid Language Adaptation," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[14] Sibo Tong, Philip N. Garner, and Herv Bourlard, "An Investigation of Deep Neural Networks for Multilingual Speech Recognition Training and Adaptation," in *Proc. Interspeech*, 2017.

[15] D. Chen and B. K. W. Mak, "Multitask Learning of Deep Neural Networks for Low-Resource Speech Recognition," *Proc. IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 7, 2015.

[16] Zoltán Tüske, Joel Pinto, Daniel Willett, and Ralf Schlüter, "Investigation on Cross-and Multilingual MLP Features Under Matched and Mismatched Acoustical Conditions," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[17] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[18] Rohit Prabhavalkar, Kanishka Rao, Tara N. Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A Comparison of Sequence-to-Sequence Models for Speech Recognition," in *Proc. Interspeech*, 2017.

[19] Hasim Sak, Andrew W. Senior, Kanishka Rao, and Françoise Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition," in *Proc. Interspeech*, 2015.

[20] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, Nov. 1997.

[21] Mike Schuster, Kuldip K. Paliwal, and A. General, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, 1997.

[22] Rich Caruana, "Multitask Learning," *Machine Learning*, 1997.

[23] Alex Graves, "Practical Variational Inference for Neural Networks," in *Proc. Neural Information Processing Systems (NIPS)*, 2011.

[24] Martín Abadi et al, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[25] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng, "Large Scale Distributed Deep Networks," in *Proc. Neural Information Processing Systems (NIPS)*, 2012.

[26] Prahallad Lavanya, Prahallad Kishore, and Ganapa Thiraju Madhavi, "A Simple Approach for Building Transliteration Editors for Indian Languages," *Journal of Zhejiang University-SCIENCE A*, vol. 6, no. 11, Nov 2005.

[27] Jan Chorowski and Navdeep Jaitly, "Towards Better Decoding and Language Model Integration in Sequence to Sequence Models," in *Proc. Interspeech*, 2017.