ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9413952

MULTI-DIALECT SPEECH RECOGNITION IN ENGLISH USING ATTENTION ON ENSEMBLE OF EXPERTS

Amit Das, Kshitiz Kumar, Jian Wu

Microsoft Speech and Language Group

{amitdas, kshitiz.kumar, jianwu}@microsoft.com

ABSTRACT

In the presence of a wide variety of dialects, training dialect-specific models for each dialect is a demanding task. Previous studies have explored training a single model that is robust across multiple dialects. These studies have used either multi-condition training, multi-task learning, end-to-end modeling, or ensemble modeling. In this study, we further explore using a single model for multi-dialect speech recognition using ensemble modeling. First, we build an ensemble of dialect-specific models (or experts). Then we linearly combine the outputs of the experts using attention weights generated by a long short-term memory (LSTM) network. For comparison purposes, we train a model that jointly learns to recognize and classify dialects using multi-task learning and a second model using multi-condition training. We train all of these models with about 60,000 hours of speech data collected in American English, Canadian English, British English, and Australian English. Experimental results reveal that our best proposed model achieved an average 4.74% word error rate reduction (WERR) compared to the strong baseline model.

***Index Terms*—** multi-dialect, attention, mixture of experts, acoustic modeling, speech recognition

1. INTRODUCTION

Recent progress in automatic speech recognition (ASR) modeling [1] can be attributed to the advances in deep neural networks (DNN) [2], convolutional neural networks (CNN) [3], and recurrent neural networks (RNN) [4,5]. However, one fundamental limitation of ASR models is that the joint distribution of the acoustic features and their corresponding labels in the training data must match with the joint distribution of the test data. Otherwise, they tend to perform poorly [6]. Thus, domain-specific models have to be built individually for each domain. Due to the presence of a wide variety of domains, real-world deployment of ASR models for each domain is usually a cumbersome procedure.

Domain variations arise due to differences in environmental conditions, speakers, languages, or dialects. Dialects are variations of a language that arise due to differences in geographical regions or social groups [7]. These variations are defined in terms of linguistic entities such as phonology, grammar, orthography, and/or vocabulary. For example, Castilian Spanish, Latin American Spanish, Rioplatense Spanish, and Caribbean Spanish are dialects of the Spanish language. Similarly, American English (EN-US), Canadian English (EN-CA), British English (EN-GB), and Australian English (EN-AU) are dialects of the English language. In this study, we explore improving the robustness of acoustic models (AMs) due to dialectal variations in English across EN-US, EN-CA, EN-GB, and EN-AU dialects.

Several studies have explored increasing the robustness of AMs across multiple dialects. Broadly, these can be classified into three categories based on their modeling approaches - (a) Adaptation [8–12]. (b) Unified [13–15], and (c) Ensemble [16–21].

In the adaptation approach, usually the parameters of a pre-trained model are fine-tuned using dialect-specific data and sometimes regularized to avoid overfitting. In [8], dialect independent Gaussian mixture models (GMMs) were adapted using dialect-specific data using maximum a posteriori (MAP) adaptation. In [9–12], multi-task learning [22, 23] was used to train shared hidden layers of a DNN [24] using data from multiple dialects. Only the top layer was adapted using dialect-specific adaptation data. While [9] and [11] used Kullback-Leibler divergence (KLD) and connectionist temporal classification (CTC) regularization respectively to avoid overfitting, [10] used i-vectors [25] as auxiliary features, and [12] used an accent classifier as an auxiliary task.

In the unified approach, labels from multiple dialects are unified to generate a single set of labels that can span across multiple dialects. In [13], a hierarchical CTC model for English dialects was trained using multi-task learning with graphemes at the primary task and dialect-specific phonemes at auxiliary tasks. The advantage of using grapheme units is that these are easier to share across dialects than phonemes. In [15], a single sequence-to-sequence (S2S) model was trained using English graphemes using data across seven English dialects. In [14], a phoneme-based model to recognize seven English dialects was trained using a single softmax layer as the primary task and a dialect classifier as the secondary task. Utterance-level dialect embeddings, extracted from a separate network, were used as auxiliary features to further improve the recognition accuracy.

In the ensemble modeling approach, several dialect-specific models (experts) are used to produce a single output. This is achieved either by selecting a particular expert or by combining the outputs of several experts. In the past, ensemble modeling has been used to improve multi-speaker phoneme recognition [16], dialect recognition using GMM-based acoustic modeling [17, 18] or hypotheses fusion [19], and environmental robustness [20, 21].

In this study, we propose an ensemble style modeling technique which combines the hidden outputs of each dialect-specific model (expert) using a small gating network (mixer) which is an LSTM model [4]. The gating network maps the hidden outputs of the experts to a probability distribution over the experts which we refer to as attention weights. These weights determine the degree of relevancy between the input features and expert outputs. The expert outputs are linearly combined using the attention weights to produce a single hidden output for recognition. Our proposed model is based on [26], also known as Recurrent Adaptive Mixture Model (RADMM), which has yielded promising results in domain robust language modeling.

2. DIALECT INFORMATION FOR FEATURE AUGMENTATION

In this section, in parallel to our proposed work, we explore training a single model with dialect ID information [27] that can be applied across all English dialects.

2.1. Augmenting input features with one-hot dialect ID

Recent studies in using one-hot language/dialect vector [12, 14, 15, 28] have showed promising gains. In our work, we created 3-dim one-hot vectors to represent the 3 dialects in EN-US, EN-GB and EN-AU. We used the EN-US dialect ID to represent both EN-US and EN-CA input data since they are mutually close. This work primarily tests the limits of our ASR systems and is not directly applicable to practical ASR systems which may not use explicit hard dialect ID information.

2.2. Augmenting input features with soft dialect ID

Here, we trained a small DNN separately as a dialect classifier to classify the three dialects using softmax predictions. Then we augmented this softmax prediction vector to the input features and trained a 4-layer LSTM model for recognition. Understandably, this work does not require explicit ground truth one-hot dialect ID vectors. Our dialect ID prediction model operates at the frame level and provides a per-frame prediction of dialect ID; we chose the design so that it does not impact the ASR latency. We refer to this model as the dialect classification and recognition (CAR) model since it predicts both the dialect type (classifier) and senones (recognizer).

2.3. Multi-task Learning

We trained another model using multi-task learning (MTL) to jointly predict the senones (ASR states) and the dialect ID by sharing all the hidden layers [12]. During backpropagation, dialect ID prediction errors embed dialect-awareness to the model thereby improving the robustness across dialects. Similar to the CAR model, this model does not require one-hot dialect ID information.

We conducted a thorough evaluation of these models but found limited or no gains from above. In particular, contrary to expectations, we did not observe any significant gains from the model which uses the 1-hot dialect vector. We can perhaps rationalize this in the context of large scale ASR training with robust acoustic model structures. Furthermore, a fraction of our training data, in particular for EN-US, may already include multi-dialectal data due to the presence of speakers from a wide variety of dialects in the US. Hence, the training corpus for EN-US may not be clearly separated among dialects. During evaluation, the CAR/MTL models showed only marginal gains (1% relative, refer Table 3). However, this work still provided us valuable insights for building ensemble models described in the next section.

3. ENSEMBLE OF EXPERTS

Our ensemble models are based on the RADMM network. The RADMM network consists of the following components - K dialect dependent models (or experts), a mixer LSTM (gating network), and a softmax output layer. In the following sections, we describe the basic RADMM model architecture [26], which we denote as RADMM-1, and our proposed improvements (RADMM-2, 3, 4). These models are illustrated in Fig. 1.

3.1. RADMM-1

An input feature vector $\mathbf{x}_t$ at time t is simultaneously fed to K experts. The k^{th} expert is identified by LSTM_k where $k = 1, \cdots, K$. Feedforwarding $\mathbf{x}_t$ through LSTM_k results in

$$(\mathbf{h}_t^{(k)}, \mathbf{c}_t^{(k)}) = \text{LSTM}_k(\mathbf{x}_t, \mathbf{h}_{t-1}^{(k)}, \mathbf{c}_{t-1}^{(k)}), \quad k = 1, \cdots, K \tag{1}$$

where $\mathbf{h}_t^{(k)}$ and $\mathbf{c}_t^{(k)}$ are the hidden output and the cell state respectively of the top-most layer of LSTM_k at time t. The same input $\mathbf{x}_t$ is fed to a mixer LSTM (LSTM_{mix}) with a projection layer followed by softmax activation. This generates the attention weight vector $\boldsymbol{\alpha}_t$ given by

$$(\mathbf{h}_t^{(mix)}, \mathbf{c}_t^{(\text{mix})}) = \text{LSTM}_{\text{mix}}(\mathbf{x}_t, \mathbf{h}_{t-1}^{(\text{mix})}, \mathbf{c}_{t-1}^{(\text{mix})}), \tag{2}$$

$$\boldsymbol{\alpha}_t = \text{softmax}(\mathbf{W}_{\text{mix}}\mathbf{h}_t^{(\text{mix})} + \mathbf{b}_{\text{mix}}), \tag{3}$$

where $\mathbf{h}_t^{(mix)}$ and $\mathbf{c}_t^{(\text{mix})}$ are the hidden output and cell state of the mixer LSTM respectively. The weight matrix $\mathbf{W}_{\text{mix}}$ projects the dimension of $\mathbf{h}_t^{(mix)}$ to K. The attention weight vector $\boldsymbol{\alpha}_t = [\alpha_t^{(1)} \cdots \alpha_t^{(K)}]$ is a probability distribution and hence $\sum_{k=1}^{K} \alpha_t^{(k)} = 1$. The weight $\alpha_t^{(k)}$ determines the importance of the k^{th} expert in producing the unified output $\mathbf{s}_t$ using

$$\mathbf{s}_t = \sum_{k=1}^{K} \alpha_t^{(k)} \mathbf{h}_t^{(k)}. \tag{4}$$

Passing $\mathbf{s}_t$ through a fully connected layer produces $\mathbf{z}_t$ which after softmax activation results in the final label posterior vector

$$\mathbf{y}_t = p(\mathbf{l}|\mathbf{x}_t) = \text{softmax}(\mathbf{z}_t), \tag{5}$$

where $\mathbf{z}_t = \mathbf{W}_o\mathbf{s}_t + \mathbf{b}_o$ is the vector of logits, $\mathbf{l}$ is the vector of labels and $(\mathbf{W}_o, \mathbf{b}_o)$ are the weight and bias parameters of the softmax layer.

3.2. RADMM-2

In RADMM-1, the mixer LSTM produces attention weights based on the input feature $\mathbf{x}_t$. Thus, it is agnostic to the state of the individual experts. To make it expert-aware, hidden outputs from the experts are fed to the mixer LSTM. This can be achieved by stacking the hidden outputs of each expert into a single column vector and feeding the stacked vector $\widetilde{\mathbf{h}}_t$ to the mixer LSTM. Thus, Eq. (2) can be modified as

$$(\mathbf{h}_t^{(\text{mix})}, \mathbf{c}_t^{(\text{mix})}) = \text{LSTM}_{\text{mix}}(\widetilde{\mathbf{h}}_t, \mathbf{h}_{t-1}^{(\text{mix})}, \mathbf{c}_{t-1}^{(\text{mix})}), \tag{6}$$

$$\text{where,} \quad \widetilde{\mathbf{h}}_t = [\mathbf{h}_t^{(1)^{\text{T}}} \mathbf{h}_t^{(2)^{\text{T}}} \cdots \mathbf{h}_t^{(K)^{\text{T}}}]^{\text{T}}.$$

With $\mathbf{h}_t^{(\text{mix})}$ computed from Eq. (6), attention weights and final softmax output can be determined using Eq. (3)-(5).

3.3. RADMM-3

In RADMM-1 and RADMM-2, the hidden outputs from each expert (i.e., $\mathbf{h}_t^{(k)}$) reside in different subspaces because each expert induces a different non-linear transformation on the input feature $\mathbf{x}_t$. By projecting the hidden outputs of the experts to a common subspace, the mixer LSTM is better able to predict the relevancy of experts with input $\mathbf{x}_t$. This is given by

$$\mathbf{g}_t^{(k)} = \mathbf{W}_k\mathbf{h}_t^{(k)}, \quad k = 1, \cdots, K. \tag{7}$$

The projected features can now be stacked as a single column to form $\widetilde{\mathbf{g}}_t = [\mathbf{g}_t^{(1)^{\text{T}}} \mathbf{g}_t^{(2)^{\text{T}}} \cdots \mathbf{g}_t^{(K)^{\text{T}}}]^{\text{T}}$ as in RADMM-2 and fed to the LSTM mixer as

$$(\mathbf{h}_t^{(\text{mix})}, \mathbf{c}_t^{(\text{mix})}) = \text{LSTM}_{\text{mix}}(\widetilde{\mathbf{g}}_t, \mathbf{h}_{t-1}^{(\text{mix})}, \mathbf{c}_{t-1}^{(\text{mix})}). \tag{8}$$

Following this, the expert weights are determined as usual using Eq. (3). Next, $\mathbf{g}_t^{(k)}$ (instead of $\mathbf{h}_t^{(k)}$) are weighted and combined using

$$\mathbf{s}_t = \sum_{k=1}^{K} \alpha_t^{(k)} \mathbf{g}_t^{(k)}. \tag{9}$$

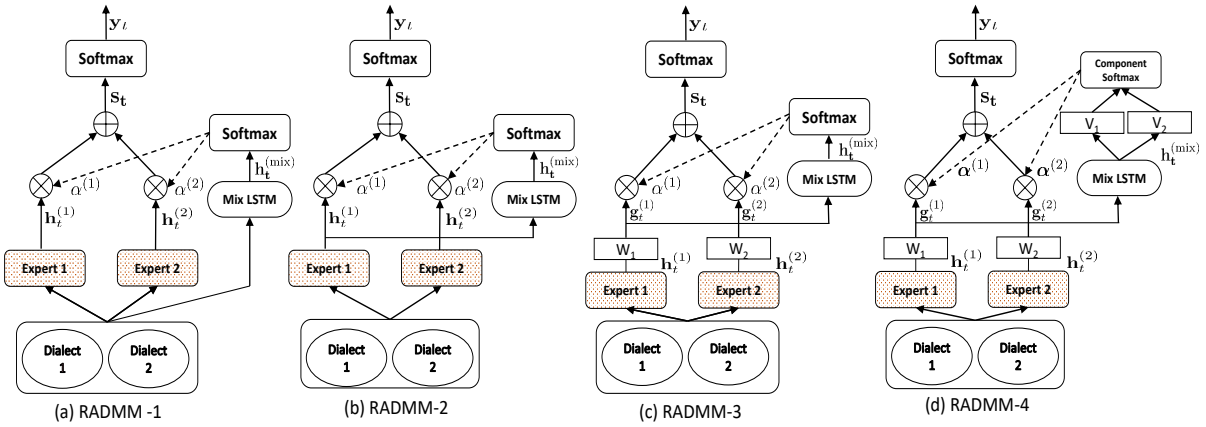

Fig. 1. Architecture of the RADMM models with $K = 2$ experts. Each expert is a deep LSTM model whereas the mixer LSTM is a shallow LSTM model. Dotted pattern in experts indicates that their parameters are not changed while training the RADMM model.

3.4. RADMM-4

By using scalar weight $\alpha_t^{(k)}$ in the linear combination in Eq.(9), all components of $\mathbf{g}_t^{(k)}$ are weighted by the same value. However, it is possible to assign a different weight to each component. Hence, we introduce component-wise (or element-wise) weighting of the expert outputs. This means instead of using scalar weight $\alpha_t^{(k)}$ for each expert, we use a vector weight $\boldsymbol{\alpha}_t^{(k)}$. This can be attained as follows. Assume $\mathbf{h}_t^{(\text{mix})}$ is available from the mixer LSTM. A scoring vector $\mathbf{e}_t^{(k)} \in \mathbb{R}^J$ is generated for each expert after passing $\mathbf{h}_t^{(\text{mix})}$ through an affine transform $(\mathbf{V}_k, \mathbf{b}_k)$ using

$$\mathbf{e}_t^{(k)} = \mathbf{V}_k \mathbf{h}_t^{(\text{mix})} + \mathbf{b}_k, \quad k = 1, \cdots, K, \tag{10}$$

A $J \times K$ scoring matrix $\mathbf{E}$ is then constructed by stacking the scoring vectors column-wise,

$$\mathbf{E} = \begin{bmatrix} | & | & & | \\ \mathbf{e}_t^{(1)} & \mathbf{e}_t^{(2)} & \cdots & \mathbf{e}_t^{(K)} \\ | & | & & | \end{bmatrix}_{J \times K}. \tag{11}$$

Keeping the j^{th} row fixed in $\mathbf{E}$, attention weights are computed by computing softmax for each row (i.e., softmax across experts). Thus, for the k^{th} expert and the j^{th} component, the attention weight $\alpha_t^{(k)}(j)$ is computed using

$$\alpha_t^{(k)}(j) = \frac{\exp(e_t^{(k)}(j))}{\sum_{k'=1}^{K} \exp(e_t^{(k')}(j))}, \quad j = 1, \cdots, J, \tag{12}$$

where $\sum_{k=1}^{K} \alpha_t^{(k)}(j) = 1, \forall j \in \{1, \cdots, J\}$. Now the projected features $\mathbf{g}_t^{(k)}$ are linearly combined component-wise with the weight vector $\boldsymbol{\alpha}_t^{(k)}$ as

$$\mathbf{s}_t = \sum_{k=1}^{K} \boldsymbol{\alpha}_t^{(k)} \odot \mathbf{g}_t^{(k)}, \tag{13}$$

where $\odot$ is the Hadamard product.

4. EXPERIMENTS AND RESULTS

Our multi-dialect English corpus includes transcribed data collected in four English dialects (Table 1) and anonymized with personally identifiable information removed. The corpus was split into two parts - train (60 k hours) and test (127 hours). The data were collected over a wide variety of acoustic conditions on Microsoft devices. These include speech collected from Cortana voice assistant, dictation, and call center conversations. We first describe the architecture of the three baseline models - the EN-US model and the dialect dependent and dialect independent models.

Table 1. English multi-dialect test corpus: Number of hours of speech data along with the number of words. (k = thousands)

English Dialect	Train	Test	
	Hours	Hours	Words
American English (EN-US)	40 k	-	-
Canadian English (EN-CA)	2.72 k	19	74 k
British English (EN-GB)	11.7 k	53	254 k
Australian English (EN-AU)	5.26 k	55	200 k
Total	59.7 k	127	528 k

4.1. EN-US Model

The EN-US model was randomly initialized and trained only with EN-US data (40 k hours). This is a latency-controlled bi-directional LSTM (LC-BLSTM) model [29]. It uses 6 hidden layers with each layer containing 1024 cells. The output of each layer is reduced to 512 dimensions using a linear projection. Thus, after concatenating the 512-dimensional projections from each forward and backward LSTM, the output of each layer is a 1024-dimensional vector. We use chunk sizes of 40 frames with 20 past and future frames. Additionally, each layer applies singular value decomposition (SVD) [30] on both the current output from the previous layer and the previous output (recurrent) from the current layer. The softmax layer has 9404 nodes to model the senone labels. We use the same set of labels across all dialects. The model has about 60 million parameters and was trained with the cross-entropy (CE) criterion using 40 k hours of EN-US data. Input features are 160-dimensional log Mel filterbank features, extracted every 10 milliseconds (ms) using 25 ms window. To reduce the run-time cost, we apply frame skipping by a factor of 2 (thereby needing only one frame to be input every 20 ms instead of two) [31].

The results are outlined in Table 2 (1st row). The "Avg." column highlights the average of WERs of EN-AU, EN-CA, and EN-GB dialects weighted by the number of test words in each dialect (# test words given in Table 1). The average WER of the EN-US model when tested across EN-AU, EN-CA, and EN-GB dialects was 14.66%.

Table 2. Baselines: WERs of EN-US, dialect dependent (DD) models and multi-condition trained dialect independent model (DI). "Avg." is computed by taking the word-weighted average of WERs measured across all EN-AU, EN-CA, EN-GB test sets.

Train Data (Model Type)	Seed	Test Data			Avg
		EN-AU	EN-CA	EN-GB	
EN-US (DD)	Rnd	13.49	9.86	16.98	14.66
EN-AU (DD)	EN-US	10.29	-	-	-
EN-CA (DD)	EN-US	-	10.76	-	-
EN-GB (DD)	EN-US	-	-	14.05	-
All (DI)	EN-US	10.78	9.85	13.65	12.03

4.2. Dialect Dependent Model (DD)

DD models were trained using the EN-US model as seed and then fine-tuned using dialect specific data. For e.g., the EN-AU DD model used the EN-US model as seed and was trained with EN-AU data alone. During decoding, dialect dependent language models (LMs) were used. For decoding the EN-CA dialect, we reused the EN-US LM because these two dialects are closely related to each other. The results are outlined in Table 2 (2nd, 3rd, 4th rows). The EN-AU and EN-GB DD models outperformed the EN-US model in their respective dialects. However, we observed regression for EN-CA dialect (9.86 → 10.76) perhaps because it had the least amount of dialect specific data compared to the other dialects. Overall, the DD models achieved an average WER of 12.16%. Compared to the EN-US model, the DD models achieved a relative WERR of 17.05% (14.66 → 12.16).

4.3. Dialect Independent Model (DI)

The DI is a dialect independent model and was trained using the EN-US model as seed and then fine-tuned after *pooling all 60 k hours* of data (multi-condition training). The WER of this model is outlined in Table 2 (last row). The average relative improvement in WER of the DI model over the EN-US model was 17.94% which is slightly better than the DD models. This is because sharing of data across multiple dialects in DI facilitates better transfer learning. Since the DI model is the best baseline model in Table 2, we use this model as our baseline model from this point onward.

4.4. Ensemble and CAR Models

In this section, we describe the ensemble (RADMM) and CAR models. We pooled speech data from all the dialects and trained the CAR model (Sec 2) and the RADMM models (Sec 3). In Table 3, we compare these models with the DI model since this is our best baseline model. To avoid any bias due to the amount of data, all models in Table 3 were trained with the same amount of data.

For our ensemble models, we first need to choose the expert models. We chose the DI and two DD models (EN-AU DD and EN-GB DD) as our experts. We found that excluding EN-CA and EN-US models did not affect the performance significantly. First, we weighted the softmax outputs of each expert uniformly (i.e. assigning $\frac{1}{3}$ weight to each expert). The WER of this model is outlined in the CWS (constant weighted softmax) row in Table 3. Constant weighting has limited merit since the weights remain constant irrespective of the dialect in the input speech. Next, because of the inclusion of dialect ID information, the CAR model performed slightly better than DI across all dialects. Next, we trained the RADMM models. The mixer LSTM is a single layer uni-directional LSTM with 512 cells using 64-dimensional linear projection. We observed that it is crucial to freeze the parameters of the experts (indicated by the dotted pattern in Fig. 1) while training the RADMM models and update the parameters of only the mixer LSTM and softmax layer. It is clear that RADMM-1 improves over the CAR model. In RADMM-2, inclusion of state information from the experts yields additional improvement over RADMM-1. In RADMM-3, using linear projections to project the hidden outputs to a common subspace improves it further. Finally, in RADMM-4, component-wise weighting (instead of scalar weighting) offers richer state combination. Comparing RADMM-4 (the best RADMM model) with the DI model, we observed WERR by about 4.74% (12.03 → 11.46). A WERR of 4.74% is significant since this translates to correctly recognizing 3000 additional words over a strong baseline model (DI). Improvements can be mostly attributed to the two resource rich dialects (EN-AU, EN-GB) while still not regressing in the EN-CA dialect. This also provides some insights about the expected improvement in performance with training data scalability. We expect higher WERR with more data per dialect. In order to decrease the size of the RADMM-4 model by half, we shared the bottom five layers of DI across all experts (similar to [9–11]) and used a sixth dialect-specific layer for each expert (DI, EN-AU, and EN-GB). We denote this as RADMM-4s (RADMM-4 shared). Although, it did not outperform the RADMM-4 model, it still achieved 3.24% WERR over the DI model using only half as many parameters in the RADMM-4 model. This makes the RADMM-4s model feasible for real-world applications.

Table 3. WERs of DI, CAR and the ensemble models (CWS, RADMM-$*$). RADMM-4s indicates RADMM-4 model with five shared hidden layers among experts.

Model	EN-AU	EN-CA	EN-GB	Avg.	WERR
DI	10.78	9.85	13.65	12.03	0
CWS	10.43	10.39	13.69	11.98	0.42
CAR	10.71	9.84	13.55	11.95	0.67
RADMM-1	10.37	9.99	13.57	11.84	1.58
RADMM-2	10.15	9.98	13.46	11.70	2.74
RADMM-3	10.08	9.95	13.06	11.50	4.41
RADMM-4	10.08	9.84	13.05	11.46	4.74
RADMM-4s	9.95	9.49	13.61	11.64	3.24

Table 4. WERs of DI and RADMM-4 models for unseen dialects. (EN-PH (Philippines), EN-PL (Poland), EN-SG (Singapore))

Dialect	Words	DI	RADMM-4	WERR
EN-PH	73372	6.48	6.20	4.32
EN-PL	82076	7.09	6.76	4.65
EN-SG	62817	12.84	12.48	2.80

In Table 4, we evaluate the performance of DI and RADMM-4 models across three unseen English dialects. None of these dialects were part of the training set. It is clear that RADMM-4 consistently outperformed DI in each of the unseen dialect and yielded WERRs in the range 2.80%-4.65%. This further demonstrates that our proposed method is not tightly coupled to seen data only. It works for unseen scenarios as well. Future direction includes increasing generalization capability of the proposed model for under-resourced seen and unseen dialects.

5. CONCLUSION

In this study, we build a single model for recognizing speech across multiple dialects in English. In particular, we build an ensemble of expert models and combine the hidden outputs of the experts by applying attention weights generated by a mixer LSTM network. We also train a model that simultaneously performs dialect recognition and classification. Experiments reveal that our best proposed ensemble model outperformed a strong dialect independent multi-condition baseline model by an average WERR 4.74%.

6. REFERENCES

[1] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA J. of Autom. Sinica.*, vol. 4, no. 3, pp. 399–412, July 2017.

[2] George E. Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan 2012.

[3] O. Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[5] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012, pp. 4085–4088.

[6] D. Yu, M. L. Seltzer, J. Li, J-T. Huang, and F. Seide, "Feature learning in deep neural networks - studies on speech recognition tasks," in *Proc. Int. Conf. Learn. Rep. (ICLR)*, 2013.

[7] R. Wardhaugh and J. Fuller, *An Introduction To Sociolinguistics*, Wiley Blackwell Publishers, 7^{th}*edition*, 2015, *IS BN*978 − 1 − 118 − 73229 − 8.

[8] D. Vergyri, L. Lamel, and J-L. Guavain, "Automatic speech recognition of multiple accented English data," in *Proc. Interspeech*, 2010, pp. 1652–1655.

[9] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the KLD-regularized model adaptation," in *Proc. Interspeech*, 2014, pp. 2977–2981.

[10] M. Chen, Z. Yang, J. Liang, Y. Li, and W. Liu, "Improving deep neural networks based multi-accent Mandarin speech recognition using i-vectors and accent-specific top layer," in *Proc. Interspeech*, 2015, pp. 3620–3624.

[11] J. Yi, Z. Wen, J. Tao, H. Ni, and B. Liu, "CTC regularized model adaptation for improving LSTM RNN based multi-accent Mandarin speech recognition," *Journal of Sig. Process. Sys.*, vol. 90, 2017.

[12] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, "Joint modeling of accents and acoustics for multi-accent speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 1–5.

[13] K. Rao and H. Sak, "Multi-accent speech recognition with hierarchical grapheme based models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2017, pp. 4815–4819.

[14] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning," in *Proc. Interspeech*, 2018, pp. 2454–2458.

[15] B. Li, T. Sainath, K. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 4749–4753.

[16] J. B. Hampshire and A. Waibel, "The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 751–769, July 1992.

[17] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S-Y Yoon, "Accent detection and speech recognition for Shanghai-accented Mandarin," in *Proc. Interspeech*, 2005, pp. 217–220.

[18] M. Elfeky, P. Moreno, and V. Soto, "Multi-dialectical languages effect on speech recognition: Too much choice can hurt," in *Proc. Int. Conf. Nat. Lang. and Speech Process. (ICNLP)*, 2015.

[19] V. Soto, O. Siohan, M. Elfeky, and Pedro J. Moreno, "Selection and combination of hypotheses for dialectal speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016, pp. 5845–5849.

[20] A. Das, J. Li, C. Liu, and Y. Gong, "Universal acoustic modeling using neural mixture models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2019, pp. 5681–5685.

[21] K. Kumar and Y. Gong, "Static and dynamic state predictions for acoustic model combination," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2019, pp. 2782–2786.

[22] Rich Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997.

[23] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 6965–6969.

[24] J-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross language knowlege transfer using multilingual deep neural network with shared hidden layers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 7304–7308.

[25] O. Glembek, L. Burget, P. Matejka, M. Karafiat, and P. Kenny, "Simplification and optimization of i-vector extraction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 4516–4519.

[26] K. Irie, S. Kumar, M. Nirschl, and H. Liao, "RADMM: Recurrent adaptive mixture model with applications to domain robust language modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 6079–6083.

[27] Tanja Schultz and Katrin Kirchhoff, *Multilingual speech processing*, Elsevier, 2006.

[28] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2019, pp. 5621–5625.

[29] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016, pp. 5755–5759.

[30] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *Proc. Interspeech*, 2013, pp. 2365–2369.

[31] Y. Miao, J. Li, Y. Wang, S. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016, pp. 2284–2288.