EXPLORATION OF LANGUAGE-SPECIFIC SELF-ATTENTION PARAMETERS FOR MULTILINGUAL END-TO-END SPEECH RECOGNITION

Brady Houston, Katrin Kirchhoff

AWS AI Labs

ABSTRACT

In the last several years, end-to-end (E2E) ASR models have mostly surpassed the performance of hybrid ASR models. E2E is particularly well suited to multilingual approaches because it doesn't require language-specific phone alignments for training. Recent work has improved multilingual E2E modeling over naive data pooling on up to several dozen languages by using both language-specific and language-universal model parameters, as well as providing information about the language being presented to the network. Complementary to previous work we analyze language-specific parameters in the attention mechanism of Conformer-based encoder models. We show that using language-specific parameters in the attention mechanism can improve performance across six languages by up to 12% compared to standard multilingual baselines and up to 36% compared to monolingual baselines, without requiring any additional parameters during monolingual inference nor fine-tuning.

Index Terms— Multilingual, CTC, self-attention

1. INTRODUCTION

The last several years have seen a seismic shift in automatic speech recognition (ASR) as the field has moved from multi-component hybrid models that utilize a phoneme-based acoustic model with a language model, lexicon and HMM decoder to all-neural (end-to-end) models that can map speech (or speech features) directly to text. Besides being less complicated (due to fewer independently trained components), end-to-end models have surpassed hybrid models in terms of overall performance in many areas [1][2].

One of the applications that has greatly benefited from the shift from a hybrid to an end-to-end approach is multilingual modeling. Previously, multilingual hybrid models required linguistic expertise to map each language's unique sets of phonemes to a canonical superset to allow for pooled training [3][4], or complicated architectures that utilized different output layers for each language, for example [5]. Due to the text-based targets typically being used for training modern end-to-end ASR systems, multilingual models can be built by simply pooling all data and ensuring that each language's alphabet is represented in the inventory of text-based tokens [6]. This greatly simplifies multilingual training, and as such, this area of research has made impressive strides as its focus has shifted from hybrid to end-to-end models.

In end-to-end multilingual training, there is typically an imbalance of data across languages [7], and the amount of training data is usually inversely related to the improvement seen over monolingual models trained on the same amount of data (i.e. languages with less training data see the most improvements) [8]. Thus much of the focus of multilingual research has been leveraging high-resource languages to improve performance on low-resource languages. However, high-resource languages can see degradation over their monolingual baselines [9][10], although increasing model size can overcome this degradation [10]. But, such models can be unwieldy to train and almost impossible to use for inference due to their size, particularly in a production environment. Alternatively, pooled-data models can be fine-tuned on individual languages to improve performance, but this requires $n_{lang} + 1$ rounds of training and produces as many models as there are languages [11].

A potential bridge between these two scenarios that mitigates each of their downsides is to incorporate language-specific parameters into a model during pooled-data training. This could remove the need for additional fine-tuning but also limit model size during inference by simply selecting those language-specific parameters trained using the target language (at the cost of increased model size during training, which may or may not be acceptable, depending on the use case). The self-attention mechanism, which has become state-of-the-art in many ASR models, is a prime candidate for exploring the effect of language-specific parameters due to its powerful sequence learning ability and the discrete yet specific functions of its component operations. Indeed, incorporating domain-specific and/or language-specific self-attention parameters has already been explored somewhat in the context of NMT [12][13][14][15] and ASR [16][17][15]. These works have mostly focused on domain/language-specific parameters in the entire self-attention mechanism, by using language/domain-specific attention heads or by entire language-specific encoder layers with self-attention. However, given the very different functions of the linear operations in attention, using language-specific parameters in the entire attention module or encoder layer may be parameter-inefficient and lead to unnecessarily

978-1-6654-7189-3/22/$31.00

large models during training.

In this paper, we take a systematic approach to explore the self-attention module in the context of a CTC-based, multilingual ASR model. We show that incorporating language-specific parameters into the several constituent linear transformations in self-attention can result in individual improvements of up to 12% on six languages compared to standard multilingual baselines. Importantly, there are also consistent improvements in all languages, with a mean improvement of 7% across languages containing differences in training data amounts of an order of magnitude. We also explore the dependency of this effect on the encoder layers in which the language-specific parameters are used. This work demonstrates the potential of an approach to multilingual modeling in which language-universal and language-specific parameters are trained jointly to improve performance, especially on low-resource languages, without requiring additional fine-tuning or increased model size during inference.

2. BACKGROUND

2.1. Self-attention

In the last few years, self-attention has largely replaced recurrent modules as the backbone of ASR models. Let $\mathbf{X} \in \mathbb{R}^{T, D_{in}}$ be a sequence of T, D_{in}-dimensional feature or activation vectors. Then, MHA is defined as:

$$\text{MHA} = \underset{0<i<N_h}{\text{concat}} \left[\text{att}^{(i)}\right] \mathbf{W}_O \tag{1}$$

$$\text{att}^{(i)} = \text{softmax}\left(\frac{\mathbf{Q}^{(i)}\mathbf{K}^{(i)T}}{\sqrt{d_k}}\right)\mathbf{V}^{(i)} \tag{2}$$

$$\mathbf{Q}^{(i)} = \mathbf{X}\mathbf{W}_Q^{(i)}, \mathbf{K}^{(i)} = \mathbf{X}\mathbf{W}_K^{(i)}, \mathbf{V}^{(i)} = \mathbf{X}\mathbf{W}_V^{(i)} \tag{3}$$

$\mathbf{W}_Q^{(i)} \in \mathbb{R}^{D_{in} \times d_k}$, $\mathbf{W}_K^{(i)} \in \mathbb{R}^{D_{in} \times d_k}$, $\mathbf{W}_V^{(i)} \in \mathbb{R}^{D_{in} \times D_{out}}$, $\mathbf{W}_O \in \mathbb{R}^{d_k N_h \times D_{out}}$, d_k is the query/key dimension $0 < (i) < N_h$ is the number of the N_h discrete attention heads. The output of the MHA module is a $\mathbb{R}^{T \times D_{out}}$ tensor of activations, which is usually passed to a subsequent convolutional and/or feed-forward module.

2.2. Multilingual modeling

The most basic approach to multilingual modeling is simply pooling data and training a model to recognize across all of the languages. Compared to monolingual models, this approach typically improves performance on low-resource languages and degrades performance on high-resource languages (or languages that significantly differ from others). This pooled model is often used as a seed model for language-specific transfer learning (fine-tuning), which can result in improved performance over monolingual models for even higher-resource languages, but requires as many additional iterations of model training as there are target languages. Two of the most common approaches to improving over naive data pooling are appending a onehot-encoded language-ID vector to input features [9][6] and including an auxiliary LID task at the final output layer of the encoder [11]. Both of these methods have been shown to yield some improvements, albeit typically modest and often inconsistent across languages.

2.3. Language-specific parameters in multilingual models

A number of studies have explored using language-specific parameters to improve the performance of multilingual end-to-end models. One of the most commonly explored approaches is to use lightweight "adapters", that consist of a down-projection, relu and up-projection for each language being used in training. Adapters have been shown to yield modest improvements, at the expense of a small number of additional model parameters [7][10]. More recent studies have explored combining outputs from language-specific linear layers with those from multilingual attentions layers [18] and a mixture-of-experts based approach to combine outputs from multilingual and monolingual recurrent layers [19]. In [17], outputs from language-specific and language-universal attention heads were concatenated and resulted in improved performance. However, the authors didn't examine language-specific matrices within the attention heads. Weight factorization approaches have also been used in multilingual modeling to decompose linear projections into language-universal and language-specific parameters [16][20]. This factorization approach is applied indiscriminately to all of the linear matrix-vector projections in the neural network, and while this results in improved performance, it does not specifically target the attention mechanism nor give a systematic understanding of where language-specific parameters are useful to the network.

3. APPROACH

We first systematically explore what components of the self-attention mechanism benefit from language-specific parameters, and what layers in the encoder benefit from language-specific attention parameters. These parameters include the head-specific Q, K, V matrices, and the final O matrix used after head concatenation (see equations 1-3). For each matrix, we use the one-hot language-id vector for each training example to select the language-specific matrices, and only those are updated during gradient descent. We compare these results to standard multilingual baselines including onehot language encoding, multitask LID and adapters. We also briefly explore using language-*family*-specific weight matrices, instead of language-specific weight matrices in an effort to minimize the number of additional model parameters during training.

During monolingual inference, the language's language-specific parameters for the attention matrices are selected using the language's one-hot encoding (which is also appended to the

Table 1. Data partitions (in hours)

Language	Train	Dev	Test
French	2000	20	62
English	1000	20	139
Spanish	1000	20	49
Italian	500	20	53
Arabic	500	20	50
Portuguese	100	20	51
Total	5100	120	404

model's input features). These parameters are combined with the remainder of the model's language-universal parameters to yield a language-specific model of the same size as a fully-multilingual (i.e. no language-specific matrices) model would be, without any fine-tuning. Thus, a-priori knowledge of the target language is necessary during inference. Future work may explore alternatives to this, such as predicting the input language using the multitask language-ID prediction module.

We also briefly explore a mixture-of-experts type approach, in which the V and O matrices each have an additional set of language-universal parameters (weights which are updated by gradients using training examples from all languages) whose activations are linearly combined with the output from the language-specific V or O parameters using a learned interpolation coefficient. This can be described by modifying equations 1-3 above:

$$\text{MHA} = \underset{0<i<N_h}{\text{concat}} \left[\text{att}^{(i)}\right] \left[\alpha_{la}\mathbf{W}_O^{la} + (1-\alpha_{la})\mathbf{W}_O^{uni}\right] \quad (4)$$

$$\text{att}^{(i)} = \text{softmax}\left(\frac{\mathbf{Q}^{(i)}\mathbf{K}^{(i)T}}{\sqrt{d_k}}\right) \left[\alpha_{la}\mathbf{V}^{(i)la} + (1-\alpha_{la})\mathbf{V}^{(i)uni}\right] \quad (5)$$

$$\mathbf{V}^{(i)la} = \mathbf{X}\mathbf{W}_V^{(i)la}, \mathbf{V}^{(i)uni} = \mathbf{X}\mathbf{W}_V^{(i)uni} \quad (6)$$

where $0 <= \alpha_{la} <= 1$ is a learned, language-specific interpolation coefficient, $\mathbf{W}_O^{la}$ and $\mathbf{W}_V^{(i)la}$ represent language-specific weight matrices and $\mathbf{W}_O^{uni}$ and $\mathbf{W}_V^{(i)uni}$ represent language-universal weight matrices.

4. METHODOLOGY

4.1. Data

Data from six languages were used in all experiments (Table 1). Four of the languages are from the Romance language family (Spanish, Italian, French and Portuguese). The data for each language are a mix of conversational telephony, call-center, media, news and read speech, as well as a mix of between one and several different dialects/accents for each language. Training data amounts were varied to simulate typical data imbalance, with Portuguese limited to 100 total hours of training data to represent a "low-resource" language. All together the amount of training data is 5100 hours (about 4.6M utterances), with 120 hours (111K utterances) of dev data equally partitioned across languages, and test partitions ranging from about 50 hours up to 140 hours. All of the languages share a written script, except for Arabic. Unigram word-piece modeling was used to generate a word-piece inventory of 2048 tokens from the pooled set of training utterances using the sentencepiece package (this inventory size was chosen to correspond with the size of the monolingual subword inventories in order to maintain a comparable number of model parameters in the mono/multilingual models). The only constraint set on this training was 100% character coverage.

4.2. Modeling

Input features to the ASR model are 80-dimensional log-mel filterbank energies extracted using a 25 ms window and 10 ms stride. The model itself is a 12-layer conformer-CTC encoder where each layer has an input/output size of 384, 8 separate attention heads and a 1024-dim, macaron-style feed-forward subnetwork [21]. Additionally, a convolutional frontend is used to achieve frame-level downsampling with a factor of four and relative positional encoding is used at each layer. To combat overfitting, several approaches are used: spectral augmentation, dropout throughout the network with a value of 0.1, weight decay with a value of 1e-6, intermediate CTC regularization after layer 6 (iCTC and finalCTC are averaged for the combined CTC loss). In addition to CTC loss, a single-layer transformer attention-decoder with 1024 units receives the output from the final encoder layer [22]. The CTC loss and att-decoder loss are averaged to get the final model training loss. The att-decoder is not used during inference. In total, the number of model parameters is approximately 42M. Models are optimized using ADAM with a gradient clipping value of 5.0 and bfloat16 mixed-precision, and trained using a warmup LR scheduler with a peak LR of 0.0033 after 25k warmup steps for a total of 60 epochs. All models are trained using ESPNet and pytorch on 1x8 Tesla A100 GPUs. Training takes approximately two days.

Several already-proven techniques for improving multilingual performance are implemented and compared to our novel approaches. Language information is fed into the network as a six-dimensional, one-hot encoded language vector appended to the filterbank features. Additionally, a multitask language-id prediction loss is combined with the CTC loss and att-decoder loss with a coefficient of 0.01. This segment-level LID loss is obtained by averaging the activations at the final encoder layer and passing them through a single, linear softmax layer. For comparison, we also examine language adapters: after each encoder layer, the activations for training examples which come from a given language are used to update the weights in each

Table 2. Effect of language-specific parameters in self-attention on WER (best result for each language and avg in **bold**)

Experiment	Train params	Infer params	Fr	It	Sp	En	Ar	Po	Avg	Avg rel to **
Monolingual	41.78M	41.78M	**22.30**	21.30	**21.20**	**21.93**	**48.97**	46.51	30.37	
Pooled	41.78M	41.78M	24.23	22.39	24.20	27.06	55.15	34.30	31.22	
Onehot	41.93M	41.93M	24.12	21.96	23.72	27.47	55.75	34.75	31.30	
LID	41.78M	41.78M	24.34	22.27	24.45	27.29	55.42	34.11	31.31	
**Onehot+LID	41.93M	41.93M	24.00	21.59	23.53	27.08	55.00	33.85	30.84	
Adapter(128)+onehot+LID	42.52M	42.03M	23.71	21.15	23.40	25.66	52.90	31.43	29.71	3.67%
Adapter(64)+onehot+LID	42.23M	41.98M	23.59	21.22	23.30	25.56	52.82	31.87	29.73	3.62%
Adapter(32)+onehot+LID	42.08M	41.96M	23.65	21.17	23.37	25.77	53.16	32.26	29.90	3.06%
Q+onehot+LID	50.80M	41.93M	23.86	21.44	23.43	26.24	53.84	32.70	30.25	1.91%
K+onehot+LID	50.80M	41.93M	23.76	21.24	23.32	25.86	53.91	32.80	30.15	2.25%
QK+onehot+LID	59.67M	41.93M	23.68	21.24	23.36	26.06	53.67	32.82	30.14	2.28%
V+onehot+LID	50.80M	41.93M	23.44	20.38	22.96	25.07	52.03	30.17	29.01	5.94%
O+onehot+LID	50.80M	41.93v	23.39	**20.29**	22.72	24.81	51.18	**29.80**	**28.70**	**6.95%**
VO+onehot+LID	59.67M	41.93M	23.50	20.32	22.88	25.11	51.46	30.64	28.99	6.02%
QKVO+onehot+LID	77.41M	41.93M	23.41	20.55	22.80	25.21	51.77	31.51	29.21	5.30%

language-specific adapter module:

$$\text{out} = W_l^U(\text{RELU}(W_l^D(\text{LayerNorm}_l(x)))) + x \quad (7)$$

where l denotes the language's index, and W^U and W^D are linear projections. In these experiments, the projected dimension size is varied from 32 to 128. With 6 adapter modules (one for each language) after each encoder layer, the total number of model parameters increases slightly to 42.5M. To compare to a multilingual approach without fine-tuning, the adapters are trained together with the rest of the model (as opposed to fine-tuning after training the seed model).

5. RESULTS AND DISCUSSION

5.1. Baselines

For comparison, we first trained several baseline models. These include monolingual models and multilingual models with several well-known techniques for improving multilingual performance (Table 2, rows 1-8). As has been previously shown, higher-resource languages (e.g. French, Spanish and English) and linguistically-dissimilar languages (e.g. Arabic), suffer degradation in multilingual models, when compared to monolingual baselines. Including a onehot language-ID vector or multitask LID prediction doesn't improve performance over naive data pooling, but the combination of the two does improve performance (all percent improvements in Table 2 are relative to this multilingual baseline), though not back to level of performance of the monolingual baseline. Adding language-specific adapters between encoder layers improves performance beyond the monolingual baselines.

5.2. Language-specific self-attention parameters

We systematically investigated including language-specific parameters for the four self-attention matrices, namely Q, K, V and O (see equations 1-3), and several combinations of these matrices. All experiments show improvements over the multilingual onehot+LID baseline (Table 2, rows 9-15). However, language-specific O and V matrices show the largest WER improvements, with language-specific O matrices improving over the onehot+LID baseline by an average of almost 7% across the six languages. The largest improvements are in the low-resource Portuguese language, which shows an improvement of 12% over the multilingual baseline and 36% improvement over its monolingual baseline. Importantly however, all languages, even those with more training resources, show some improvement over the multilingual baseline, with French being the smallest improvement at 2.5% and a median improvement across all six languages of 6.5%. Additionally, language-specific V and O matrices outperform language-specific adapters (Table 2, rows 6-8) by 2.4% and 3.4%, on average. Although language-specific O and V matrices yielded consistent improvements over the baseline multilingual model, these models still only outperform monolingual baselines for Italian and Portuguese. This is a well-known phenomenon; higher-resource languages typically suffer degradation when pooled with data from lower-resource languages for training a

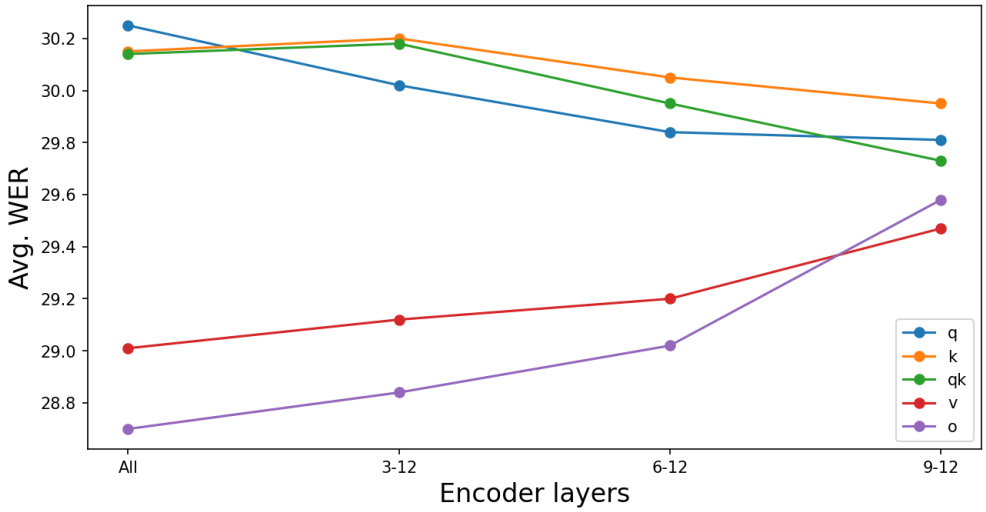

Fig. 1. Effect of language-specific parameters by encoder layers

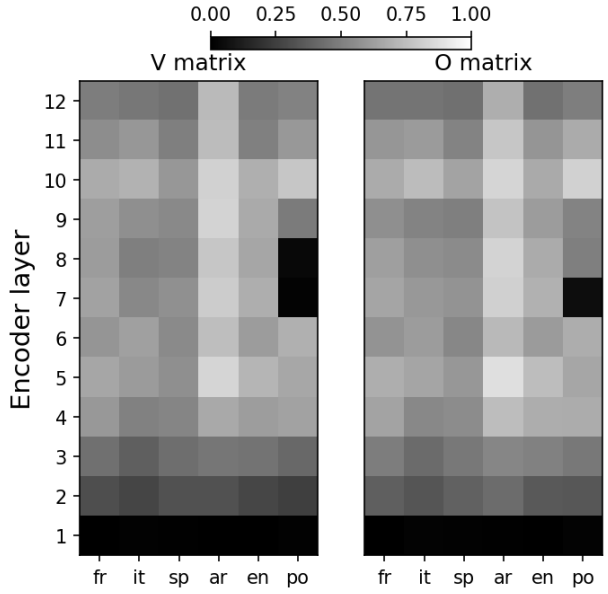

Fig. 2. Interpolation weight between language-specific and multilingual O parameters by encoder layer (higher interpolation weight means favoring language-specific parameters.

multilingual model, even when using approaches like adding a onehot-encoded language-ID vector to features, as done here.

These results intuitively make sense; because Q/K matrices are only used to generate attention scores, language-specific parameters may not be useful in these operations (at least until deeper layers where the model is likely learning higher levels of abstraction, e.g. grammar - see section below). The V and O matrices, however, benefit from language-specific parameters because they directly create/modify the activations that are passed to the subsequent convolutional/feedforward sublayers and then the following encoder layers and thus permit the transmission of language-specific information throughout the model. Interestingly, neither the combination of language-specific V and O matrices, nor using language-specific parameters for the entire self-attention layer yields better performance that the language-specific O matrix, showing that the self-attention module does indeed benefit from parameters that are language-universal (i.e. that are updated using training examples from all languages during gradient descent).

It should be noted that including language-specific parameters linearly increases the overall number of parameters in the model during training according to the number of languages (see table 2, column $Tr\#$). For example, including language-specific O matrices increases total number of parameters by about 20% for six languages. However, for monolingual inference, the language-specific parameters can be simply be selected out and combined with the rest of the language-universal model parameters to form a language-specific model with the same number of parameters as the multilingual baseline model (i.e. the onehot+LID model; see table 2, column $Inf\#$). Scaling training to many more languages may become challenging, but there are also potential trade-offs to balance language-specific parameters and the total number of training parameters. One such trade-off is to use language family-specific parameters, instead of language-specific parameters. We briefly explored this, using the same O matrix for the four Romance languages, while English and Arabic still each had their own O matrix. This decreased the number of additional parameters by half, but still gave an average performance improvement across the six languages of 4.8% (median=5.2%) over the multilingual baseline. Another such trade-off is to only use language-specific parameters in a subset of encoder layers, which is detailed in the following section.

Additionally, although the total number of training parameters increases, overall memory usage during training is not greatly impacted because the vast majority of GPU memory is used to store model activations, rather than model parameters. The activations from language-specific matrices are gathered by training example language ID before being passed to subsequent operations, and thus don't greatly increase the overall memory usage from model activations.

5.3. Mixture of experts

We briefly explored using a learned interpolation between language-specific and language-universal activations for the V and O matrices (see equations 4-6 above). Performance was slightly better than when using the language-specific parameters alone; average and median WER improvement across the six languages was 6.5% and 6.1% respectively, when using the mixed V matrix (compared to 5.9% and 5.5% for language-specific-only parameters), and 7.2% and 6.7%, respectively for the mixed O matrix (compared to 7.0% and 6.5% for language-specific-only parameters). Using the linear combination of language-specific and language-universal O activations gave a 13.3% improvement for Portuguese, the best result seen through all experiments. As expected, the interpolation weights showed dependencies on both encoder layer and language (Figure 2). The first encoder layer favored language-universal weights for all languages and subsequent layers became more focused on language-specific parameters. Across languages, Arabic typically showed the highest interpolation weight across layers (i.e. language-specific parameters favored the most). For Portuguese, layers 7-8 showed very low interpolation weights (favoring language-universal parameters). This may be explained by the intermediate CTC layer receiving the outputs from layer 6; this regularization may be forcing the model to revert to language-universal parameters for the low-resource languages.

5.4. Language-specific self-attention parameters by encoder layer

Language-specific self-attention matrices improve performance over multilingual (and averaged monolingual) baselines when used in all encoder layers. However, it is useful to understand if the effect of language-specific parameters is dependent on encoder layers. So, we applied the language-specific parameters to layers 3-12, 6-12 and 9-12 to compare to the results in Table 2 (Figure 1). Language-specific O/V and Q/K matrices show differential effects by the encoder layers in which they were applied; while O/V give biggest improvements when applied in all layers, Q/K give biggest improvements when applied at the final three encoder layers. This makes sense, as the higher layers of abstraction at deeper levels of the encoder may benefit from language-specific attention scores created by the language-specific Q/K matrices to capture phenomena like different grammar rules, while language-specific activations being passed between all layers could benefit performance through all layers of abstraction. We also briefly explored combining these effects by using language-specific O matrices in all encoder layers but language-specific Q/K matrices in only the final three encoder layers. Improvements were not better than language-specific O matrices alone (average/median improvement of 5.4% and 4.8%, respectively), suggesting that language-specific O matrices are able to utilize language-universal attention scores better than language-specific attention scores.

6. CONCLUSION

In this work, we systematically explore including language-specific parameters in the self-attention module of a multilingual model trained on about 5100 hours of data across six, imbalanced languages. We show that language-specific O and V matrices can improve performance by up to 12% compared to multilingual baselines and up to 36% compared to monolingual baselines. This work bridges a gap between very large multilingual models that can outperform monolingual baselines for even high resource languages and smaller models that require fine-tuning on all target languages. It does so by including language-specific parameters that increase model size during training, but maintains the same size at inference as would the baseline model with all language-universal parameters. Future work will continue to address this gap with the goal of creating multilingual models that maintain large improvements on low-resource languages but avoid degradation on higher-resource languages.

7. REFERENCES

[1] Jinyu Li, Rui Zhao, Zhong Meng, Yanqing Liu, Wenning Wei, Sarangarajan Parthasarathy, Vadim Mazalov, Zhenghao Wang, Lei He, Sheng Zhao, and Yifan Gong, "Developing RNN-T Models Surpassing High-Performance Hybrid Models with Customization Capability," pp. 2–6, 2020.

[2] Jinyu Li, Yu Wu, Yashesh Gaur, Chengyi Wang, Rui Zhao, and Shujie Liu, "On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition," 2020.

[3] Harish Arsikere, Ashtosh Sapru, and Sri Garimella, "Multi-dialect acoustic modeling using phone mapping and online i-vectors," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 2125–2129, 2019.

[4] Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R. Mortensen, Graham Neubig, Alan W Black, and Florian Metze, "Universal Phone Recognition with a Multilingual Allophone System," 2020.

[5] 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," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 7304–7308, 2013.

[6] Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual Speech Recognition with a Single End-to-End Model," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, pp. 4904–4908, 2018.

[7] Anjuli Kannan, Arindrima Datta, Tara N. Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 2130–2134, 2019.

[8] Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, "Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters," , no. iv, 2020.

[9] Shinji Watanabe, Takaaki Hori, and John R Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 265–271.

[10] Bo Li, Ruoming Pang, Tara N. Sainath, Anmol Gulati, Yu Zhang, James Qin, Parisa Haghani, W. Ronny Huang, Min Ma, and Junwen Bai, "Scaling End-to-End Models for Large-Scale Multilingual ASR," 2021.

[11] Wenxin Hou, Yue Dong, Bairong Zhuang, Longfei Yang, Jiatong Shi, and Takahiro Shinozaki, "Large-scale end-to-end multilingual speech recognition and language identification with multi-task learning," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2020-Octob, pp. 1037–1041, 2020.

[12] Graeme Blackwood, Miguel Ballesteros, and Todd Ward, "Multilingual neural machine translation with task-specific attention," *arXiv preprint arXiv:1806.03280*, 2018.

[13] Shiqi Zhang, Yan Liu, Deyi Xiong, Pei Zhang, and Boxing Chen, "Domain-Aware Self-Attention for Multi-Domain Neural Machine Translation," pp. 2047–2051, 2021.

[14] Junwei Liao, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng, "Improving zero-shot neural machine translation on language-specific encoders-decoders," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[15] Hongyu Gong, Yun Tang, Juan Pino, and Xian Li, "Pay better attention to attention: Head selection in multilingual and multi-domain sequence modeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2668–2681, 2021.

[16] Ngoc-Quan Pham, Tuan-Nam Nguyen, Sebastian Stüker, and Alex Waibel, "Efficient Weight Factorization for Multilingual Speech Recognition," pp. 2421–2425, 2021.

[17] Yun Zhu, Parisa Haghani, Anshuman Tripathi, Bhuvana Ramabhadran, Brian Farris, Hainan Xu, Han Lu, Hasim Sak, Isabel Leal, Neeraj Gaur, Pedro J. Moreno, and Qian Zhang, "Multilingual speech recognition with self-attention structured parameterization," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2020-October, pp. 4741–4745, 2020.

[18] Long Zhou, Jinyu Li, Eric Sun, and Shujie Liu, "A Configurable Multilingual Model is All You Need to Recognize All Languages," 2021.

[19] Neeraj Gaur, Brian Farris, Parisa Haghani, Isabel Leal, Pedro J. Moreno, Manasa Prasad, Bhuvana Ramabhadran, and Yun Zhu, "Mixture of informed experts for multilingual speech recognition," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June, pp. 6234–6238, 2021.

[20] Ngoc-Quan Pham, Alex Waibel, and Jan Niehues, "Adaptive multilingual speech recognition with pretrained models," 2022.

[21] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[22] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 4835–4839, 2017.