A CONFIGURABLE MULTILINGUAL MODEL IS ALL YOU NEED TO RECOGNIZE ALL LANGUAGES

Long Zhou[†], *Jinyu Li*[‡], *Eric Sun*[‡], *Shujie Liu*[†]

[†]Microsoft Research Asia
[‡]Microsoft Speech and Language Group

ABSTRACT

Multilingual automatic speech recognition models have shown great promise in recent years because of the simple model training and deployment process. Conventional methods either train a universal multilingual model without taking any language information or with a 1-hot language ID (LID) vector to guide the recognition of the target language. In practice, a multilingual user can be prompted to pre-select several languages he/she can speak. The multilingual model without LID cannot well utilize the language information set by the user while the multilingual model with 1-hot LID can only handle one pre-selected language. In this paper, we propose a novel configurable multilingual model (CMM) which is trained only once but can be configured as different models based on users' choices by extracting language-specific modules together with a universal module from the trained CMM. Particularly, a single CMM can be deployed to any user scenario where the users can pre-select any combination of languages. Trained with 75K hours of transcribed anonymized Microsoft multilingual data and evaluated with 10-language test sets, the proposed CMM improves from the universal multilingual model by 26.0%, 16.9%, and 10.4% relative word error reduction when the user selects 1, 2, or 3 languages, respectively.

***Index Terms*—** multilingual speech recognition, configurable multilingual model, transformer-transducer.

1. INTRODUCTION

According to [1], there are 40%, 43%, 13%, 3%, and less than 0.1% people in the world can speak 1, 2, 3, 4, and 5+ languages fluently. With the advance of deep learning [2], the commercial monolingual automatic speech recognition (ASR) systems are highly optimized with excellent accuracy [3, 4]. There are increasing interests in developing high-quality commercial ASR systems that can recognize speeches from multiple languages without letting users explicitly indicate which language he/she will speak for every utterance. A common practice in industry is described in [5] which has an interface to enable the user to select multiple languages and use a language ID (LID) detector to select the decoding output from the ASR models of all selected languages. However, this method is cost-consuming because it needs to run multiple speech recognizers at the same time, and the LID estimation usually introduces latency because it needs a period of speech in order to have reliable decisions.

In the context of end-to-end (E2E) modeling [6, 7, 8, 9, 10, 11], the easiest way is pooling the data of all languages to build a single multilingual model [12, 13]. This model is a universal model, and can recognize the speech from any language as long as this language is used during training. Although the sharing is maximized, it also brings confusion across languages which can be alleviated by taking a 1-hot LID vector as the additional input so that the multilingual model is guided to reduce the confusion from other languages and recognize that specific language well [14]. The multilingual model without LID input cannot take advantage of the user selection. In contrast, the multilingual model with 1-hot LID vector needs to know which language the user will speak in advance, and cannot work for multilingual speakers who only pre-select several languages once. Another solution is to build a specific model for every combination of languages so that we can deploy the model based on any user's selection. However, the development cost is formidable. For example, if we want to have bilingual and trilingual support of 10 languages, we have to build $C_{10}^2 = 45$ and $C_{10}^3 = 120$ specific models with such solution.

In this work, we design a *configurable multilingual model (CMM)* that can be configured to recognize speech from any combination of languages based on a multi-hot LID vector selected by users. The hidden output is calculated as the weighted combination of the output from a universal module and the outputs from all language-specific modules using the multi-hot vector. The universal module is language independent, modeling the shared information of all languages. The residue of any language from the shared one carries much less information, hence only a small number of parameters are needed to model the residue. At runtime, the universal module together with corresponding language-specific modules are activated based on user choice. For the first time, we propose a configurable multilingual model with a multi-hot LID vector. By leveraging it, we reduce the language confusion space from dozens to only a few, hence significantly improving the recognition accuracy.

CMM is very different from the dynamic language switching concept explored in [15], whose goal is to train a joint model for LID prediction and ASR under a multi-task learning framework. CMM also differs from the recent multilingual ASR models using mixture of experts (MoE) [16, 17], in which every expert has the same amount of parameters as the universal module. In contrast, CMM is only slightly larger than the universal model due to the residue modeling. The structure of CMM is also different from the structure in [18] which has a separate adapter module for every language without the universal module. Most importantly, to our best knowledge, there is no work of configuring a single model for better recognition of any combination of languages selected by multilingual users.

2. MODEL

Our goal is to design a single model which can be configured at the inference time to recognize any language combination based on user selection. This is realized with our proposed configurable multilingual model which is based on the multilingual streaming Transformer Transducer model.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

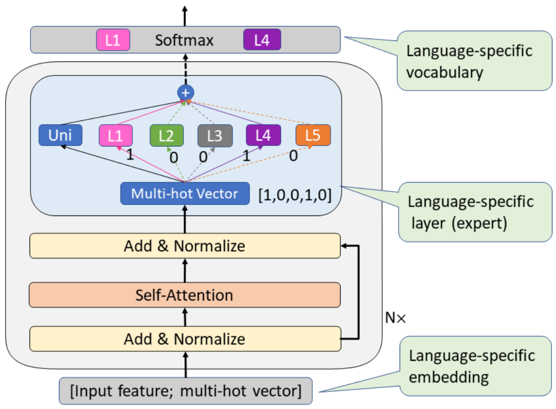

Fig. 1. Diagram of configurable multilingual model (CMM). Uni denotes universal feed-forward network, and Li denotes specific layer for language L_i. We propose language-specific embedding, layer, vocabulary to support configurable ASR.

2.1. RNN and Transformer Transducer

Because of its streaming nature, RNN-Transducer (RNN-T) [7] has become a very promising E2E model in industry to replace the traditional hybrid models [19, 20, 21]. RNN-T contains an encoder network, a prediction network, and a joint network. The encoder network converts the acoustic feature x_t into a high-level representation h_t^{enc}, where t is time index. The prediction network produces a high-level representation h_u^{pre} by conditioning on the previous non-blank target y_{u-1}, where u is the output label index. The joint network is a feed-forward network that combines the encoder network output h_t^{enc} and the prediction network output h_u^{pre} to generate $h_{t,u}$ which is used to calculate softmax output.

Given the great success of Transformer [22], Transformer Transducer (T-T) [23, 24] was proposed to replace LSTM with Transformer [22] in the encoder of Transducer with significant gain. Chen et al. [25] proposed an efficient implementation of T-T with very small latency and computation cost, while maintaining high recognition accuracy. We use the T-T model in [25] as the backbone model.

2.2. Multilingual Speech Recognition

Training a single ASR model to support multiple languages is promising and challenging [26, 14, 18, 27, 28]. Previous work has demonstrated the importance of language ID (LID) [29], with which the multilingual system can significantly outperform the universal multilingual system without LID. A simple but effective way to leverage the LID is representing the LID as a 1-hot vector, and appending it to the input layer of the encoder network. Formally, the new input acoustic feature vector x_t^{new} can be denoted as:

$$x_t^{new} = [x_t; d_l] \tag{1}$$

where $[;]$ means concatenation operation, and d_l is a 1-hot vector where the corresponding dimensionality of LID is equal to one, others are zeros, such as $[0, 0, 0, 1, 0]$.

Although the multilingual model with the 1-hot LID vector can obtain a significant improvement than a universal model without LID by taking advantage of the user selection, it needs to know which language the user will speak in advance for every utterance, and cannot work for the popular scenario where the multilingual user can speak few languages and pre-select those languages once in the interface.

2.3. Configurable Multilingual Model

To cover all the usage scenarios for multilingual users, we propose a configurable multilingual model (CMM), which also naturally supports recognizing code-switching speech. Fig. 1 shows the encoder network part of CMM. The universal module (uni) is the same as the Transformer encoder of a standard multilingual ASR system. Compared to the universal model, CMM employs **language-specific embedding**, **language-specific layer**, and **language-specific vocabulary** to achieve the highly configurable goal.

We use the multi-hot vector as the user choice vector to represent languages selected by the user and concatenate it with input acoustic feature to build a **language-specific embedding** as Eq. (1). For example, $[1, 0, 0, 1, 0]$ means that the user chooses both first and fourth languages at inference.

To further enhance the model ability of distinguishing different languages, we design a **language-specific layer** used in the encoder network or prediction network. At layer l of encoder network, we have the universal module (uni) and N language-specific modules ($\texttt{Linear}_i, i = 1...N$), where N is the total number of languages in training, as shown in Fig. 1. The layer input v is passed into every module to generate the output h_{uni} and $h_{spe,i}$.

$$h_{att}^l = \texttt{Attention}(\texttt{LayerNorm}(v^{l-1})) + v^{l-1} \tag{2}$$

$$h_{uni}^l = \texttt{FFN}(\texttt{LayerNorm}(h_{att}^l)) + h_{att}^l \tag{3}$$

$$h_{spe,i}^l = \texttt{Linear}_i(\texttt{LayerNorm}(h_{att}^l)), i = 1, 2, ..., N \tag{4}$$

where $\texttt{LayerNorm}$, $\texttt{Attention}$, and $\texttt{FFN}$ denotes layer normalization, self-attention, and feed-forward network, respectively. By combining the universal representation h_{uni}^l and specific representation $h_{spe,i}^l$, the formulation of the output v^l at the l-th layer is

$$v^l = h_{uni}^l + \sum_{i=1}^{N} w_i h_{spe,i}^l \tag{5}$$

The weight w_i is determined by the user choice vector: (1) 1-hot vector, i.e., the user selects only one language; (2) multi-hot vector, i.e., the user selects multiple languages: a vector with several 1 elements (corresponding to user choice) and the remaining elements as 0. w_i will be normalized by the total number of 1 in the vector.

We further apply the specific module into the output of the prediction network. Formally, by utilizing a feed-forward network, the joint network combines the encoder network output h_t^{enc} and the prediction network output h_u^{dec} as:

$$z_{t,u} = f^{joint}(h_t^{enc}, h_u^{dec}) \tag{6}$$

$$= \phi(Uh_t^{enc} + Vh_u^{dec} + \sum_{i=1}^{N} w_i h_{spe,i}^{dec} + b_z) \tag{7}$$

where $h_{spe,i}^{dec} = \texttt{Linear}_i(h_u^{dec})$, is the proposed language-specific prediction-network output for language L_i. U and V are weight matrices, b_z is a bias vector, and ϕ is a non-linear function.

Moreover, we design a **language-specific vocabulary** strategy. Given the vocabulary of each language $V_1, ..., V_N$ and total vocabulary V_{total}, we can merge the corresponding vocabularies of user choice to a temporary vocabulary V_{tmp} at inference. V_{tmp} is smaller than V_{total}, which can be used to avoid the generation of unexpected tokens from other languages not selected by users.

Table 1. WER of baselines and our proposed configurable multilingual model, which is trained with maximum 3 languages combinations. CMM with 1/2/3-hot LID means monolingual, bilingual, and trilingual scenario, respectively.

Model	EN	ES	FR	IT	PL	PT	NL	DE	RO	EL	AVE
Monolingual Model	9.52	19.98	21.58	19.67	17.39	14.58	20.74	16.26	14.91	17.63	17.22
Multilingual w/o 1-hot LID	10.72	19.83	27.02	21.59	23.99	14.14	24.41	18.16	15.56	17.83	19.32
Multilingual w/ 1-hot LID	10.50	16.07	17.43	15.30	13.69	13.01	17.70	16.24	14.62	17.43	15.20
CMM w/ 1-hot LID	9.90	14.82	16.68	12.57	13.73	12.26	17.23	15.44	13.72	16.57	14.29
CMM w/ 2-hot LID	10.04	15.88	19.66	14.65	18.63	12.86	20.96	16.46	14.45	16.98	16.06
CMM w/ 3-hot LID	10.14	17.04	22.35	16.60	21.63	13.40	22.80	17.18	14.85	17.20	17.32

Table 2. Number of utterances in train and test sets.

LANG	Train	Test	LANG	Train	Test
EN	32.6M	266.9K	ES	6.7M	42.6K
FR	5.9M	42.8K	PT	3.6M	21.4K
IT	6.0M	24.7K	NL	0.6M	7.9K
PL	1.4M	6.1K	DE	4.7M	49.0K
RO	1.2M	16.7K	EL	1.5M	26.0K

2.4. Model Training

The key to train CMM is that we need to simulate the combination of languages selected by users. To do that, for each training sample, we generate the user choice multi-hot vector by randomly setting several (or zero for 1-hot vector) elements together with the ground truth element as 1, and setting other elements as 0. In this way, CMM is informed that the current training sample comes from one of the several languages set by the user choice vector. During training, we go through all the combinations of languages.

We have two strategies to train CMM. The first is to train CMM from scratch. The second one is to first train the universal module using the training data without user choice vector. Then we train language-specific modules using training data with user choice vector by fine-tuning the pre-trained model. To reduce memory consumption, we only apply a language-specific linear layer to the top and bottom layers instead of all encoder network layers.

3. EXPERIMENTS

3.1. Dataset

We investigate the performance of the proposed configurable multilingual model on 75 thousand (K) hours of transcribed Microsoft data. The training set and test set cover 10 languages, including English (EN), Spanish (ES), French (FR), Italian (IT), Polish (PL), Portuguese (PT), Netherlands (NL), German (DE), Romanian (RO), and Greek (EL). The size of training data for each language varies due to the availability of transcribed data from 0.6 Million (M) utterances to 32.6M, which are shown in Table 2. All the training and test data are anonymized data with personally identifiable information removed.

3.2. Implementation Details

All experiments in this paper employ 80-dimensional log-Mel filter bank features, computed with a 25 millisecond (ms) window, and the frame shift is 10ms. Following [25], we apply a future context window of 18 and a left chunk of 4 for the input acoustic feature. We set the vocabulary as 10K sentence pieces trained on the training data transcription of all languages. In terms of Transformer Transducer, 18 transformer layers with 512 hidden units and 2048 feed-forward nodes are used as the encoder network, and 2 LSTM layers with 1024 memory cells are used as the prediction network. Finally, the joint network also has 512 hidden units. A 10-dimensional multi-hot language vector is fed into the encoder as an additional input to CMM. The numbers of parameters of multilingual w/o LID baseline, multilingual w/ 1-hot LID baseline, and CMM are 80.9M, 81.0M, and 91.5M, respectively. The Adam algorithm [30] with gradient clipping and warmup is used for optimization. We train models using 32 NVIDIA V100 GPUs, and report the word error rate (WER) for every language and also averaged WER over all languages.

3.3. Multilingual vs. monolingual models

We first compare the monolingual models and multilingual models. As shown in Fig. 1, compared to the monolingual baselines which are trained independently on the data from each language, the universal multilingual model without LID which simply concatenates training samples of all languages gets a relative 12.2% WER increase on average for all 10 languages. These results show the challenge of the universal model without knowing which language the user will speak in advance. If knowing which language the user will speak in advance, the multilingual model with 1-hot LID can achieve a 21.3% relative WER (WERR) reduction from the multilingual universal model without LID, and it outperforms monolingual baselines by 11.7% WERR over ten languages. The experimental results show the importance of leveraging user selection by taking a 1-hot LID vector as the additional input that can guide the recognition of the current language.

3.4. CMM vs. monolingual/multilingual models

The results of CMM which is trained with maximum 3 languages combinations are also shown in Table 1. Similar to the multilingual model with 1-hot LID, CMM with 1-hot LID has the same setting at inference and achieves better performance (6.0% WERR) than the multilingual model with 1-hot LID, which demonstrates that the proposed language-specific modules are beneficial for multilingual speech recognition. CMM also supports bilingual and trilingual languages decoding. During evaluation, in addition to the current language, we also randomly assign one or two other languages by constructing 2-hot and 3-hot LID vectors to simulate the cases that users select two or three languages, respectively. The results show that the performance of our proposed CMM with 2-hot LID and CMM with 3-hot LID is between the universal multilingual model without LID and the specific multilingual model with 1-hot LID, which meets our experimental expectations of supporting user's multiple

Table 3. Ablation study. The CMM w/ 2-hot LID, which is trained from scratch and decoded with 2-hot LID, is used as baseline. "- specific embedding", "- specific layer", and "- specific vocabulary" means removing the language-specific embedding, layer, and vocabulary, respectively. CMM-Finetune w/ 2-hot LID denotes the model fine-tuned from the universal model and decoded with 2-hot LID.

Model	EN	ES	FR	IT	PL	PT	NL	DE	RO	EL	AVE
CMM w/ 2-hot LID	10.04	15.88	19.66	14.65	18.63	12.86	20.96	16.46	14.45	16.98	16.06
-specific embedding	10.10	16.26	20.62	14.78	18.68	12.80	20.83	16.45	14.36	17.09	16.20
-specific layer	10.50	16.91	21.79	15.24	19.09	13.60	21.77	16.84	15.19	17.51	16.84
-specific vocabulary	10.04	15.88	19.67	14.67	18.65	12.86	20.95	16.46	14.45	16.98	16.06
-encoder specific layer	9.99	16.57	20.87	14.56	18.94	13.18	21.90	16.27	14.48	17.08	16.38
-prediction specific layer	10.52	16.66	20.68	14.79	18.79	13.26	22.10	16.92	15.08	17.59	16.64
CMM-Finetune w/ 2-hot LID	9.43	15.80	21.36	14.50	19.29	11.88	20.11	15.63	13.42	16.11	15.75

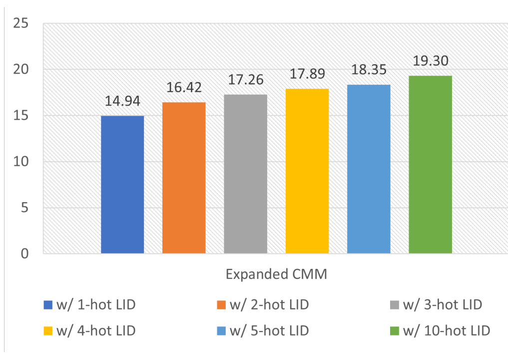

Fig. 2. Average WER of expanded configurable multilingual model with different multi-hot vector on 10 languages.

selections: supporting more languages brings more confusion which reduces the model's focus on a single language. CMM improves from the universal multilingual model by 26.0%, 16.9%, and 10.4% WERR when the user selects 1, 2, or 3 languages, respectively. For a fair comparison, we also enlarge the hidden layer size of the universal model to have a similar parameter size as the CMM. Results show that the enlarged universal model gets a slight gain than the standard universal model (18.83% vs. 19.32%), and our CMM still significantly outperforms the enlarged universal model.

3.5. Ablation study

Different from the conventional multilingual model, the proposed CMM employs three specific modules, including language-specific embedding, layer, and vocabulary, as introduced in Section 2.3. In this section, we first conduct an ablation study to analyze the effectiveness of each module.

We show in Table 3 the WER of different configurable model variants for the task of CMM with 2-hot LID. The CMM without language-specific embedding and language-specific layer obtain 16.20% and 16.84% average WER on all languages, worse than the baseline CMM by 0.8% and 4.8% relative, respectively. It demonstrates that language-specific layer is more important than language-specific embedding. Although the CMM without language-specific vocabulary obtains similar WER, employing language-specific vocabulary can avoid outputting the unexpected token of languages not selected by users, hence improving user experience.

Second, as in the previous analysis, the language-specific layer is the key component of the proposed CMM. This prompts us to further break down the contribution of the language-specific layer into the contribution from the encoder network and the prediction network. Results show that CMM without the language-specific layer of the prediction network performs worse than CMM without the language-specific layer of the encoder. Therefore, the specific layer in the prediction network is slightly more critical than the specific layer in the encoder network in CMM.

Finally, we compare two different training methods: training from scratch and fine-tuning from a universal model, as introduced in Section 2.4. As shown in Table 3, CMM is trained from scratch, and CMM-Finetune uses the same setting but it is fine-tuned from a universal model. The two models get 16.06% and 15.75% average WER, respectively, which demonstrates that the fine-tuning strategy is better than training from scratch for CMM.

3.6. Scalability of CMM

We further conduct the experiments of expanded CMM, in which any combination of 10 languages can be chosen by users. At inference, we verify the configurable model given 1-hot, 2-hot, 3-hot, 4-hot, 5-hot, or 10-hot LIDs. As shown in Figure 2, we can draw several conclusions: (1) The CMM training method is still effective when the maximum combination is expanded to all 10 language; (2) The more languages the user selects at inference, the higher the WER of the CMM; (3) CMM with 10-hot LID can recognize the same 10 languages as the universal model, both of which achieve comparable performance on all languages; (4) Due to more user choices, the frequency of each language combination in training will decrease. This is a potential reason that expanded CMM performs comparably or worse than CMM with maximum 3 languages combinations on the cases with 1-hot, 2-hot, and 3-hot LID, while CMM with maximum 3 languages combinations cannot handle the recognition with more than 3 languages selected by users.

4. CONCLUSION

In this paper, we proposed a configurable multilingual model (CMM) which consists of a universal multilingual module and a specific module for each language. CMM can be configured to recognize speeches from any combination of languages while taking advantage of user selection. More importantly, we only train CMM once but can deploy different models based on user choice by using language-specific embedding, layer, and vocabulary. Besides, CMM is only slightly larger than the universal multilingual model. Massive experiments are conducted on 75K hours of transcribed anonymized Microsoft data with 10 languages. Results demonstrate that the proposed CMM achieves a significant WER reduction from the universal multilingual model.

5. REFERENCES

[1] "Multilingual people," http://ilanguages.org/bilingual .php, Accessed: 2021-04-16.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Bo Li, Tara N Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Hasim Sak, Golan Pundak, Kean K Chin, et al., "Acoustic modeling for google home.," in *Interspeech*, 2017, pp. 399–403.

[4] Jinyu Li, Rui Zhao, Zhuo Chen, Changliang Liu, Xiong Xiao, Guoli Ye, and Yifan Gong, "Developing far-field speaker system via teacher-student learning," in *Proc. ICASSP*, 2018, pp. 5699–5703.

[5] Javier Gonzalez-Dominguez, David Eustis, Ignacio Lopez-Moreno, Andrew Senior, Françoise Beaufays, and Pedro J Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of selected topics in signal processing*, vol. 9, no. 4, pp. 749–759, 2014.

[6] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *ICML*, 2006, pp. 369–376.

[7] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[8] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*. IEEE, 2016, pp. 4960–4964.

[9] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[10] 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," in *Proc. Interspeech*, 2020.

[11] Jinyu Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, 2022.

[12] Shinji Watanabe, Takaaki Hori, and John R Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Proc. ASRU*. IEEE, 2017, pp. 265–271.

[13] Suyoun Kim and Michael L Seltzer, "Towards language-universal end-to-end speech recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 4914–4918.

[14] 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," in *Proc. ICASSP*. IEEE, 2018, pp. 4904–4908.

[15] Surabhi Punjabi, Harish Arsikere, Zeynab Raeesy, Chander Chandak, Nikhil Bhave, Ankish Bansal, Markus Müller, Sergio Murillo, Ariya Rastrow, Andreas Stolcke, et al., "Joint asr and language identification using rnn-t: An efficient approach to dynamic language switching," in *ICASSP 2021*. IEEE, 2021, pp. 7218–7222.

[16] Amit Das, Kshitiz Kumar, and Jian Wu, "Multi-dialect speech recognition in English using attention on ensemble of experts," in *Proc. ICASSP*, 2021.

[17] 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," in *Proc. ICASSP*. IEEE, 2021, pp. 6234–6238.

[18] 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," *arXiv preprint arXiv:1909.05330*, 2019.

[19] Tara N Sainath, Yanzhang He, Bo Li, et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. ICASSP*, 2020, pp. 6059–6063.

[20] Jinyu Li, , Rui Zhao, Zhong Meng, et al., "Developing RNN-T models surpassing high-performance hybrid models with customization capability," in *Proc. Interspeech*, 2020.

[21] Xiaohui Zhang, Frank Zhang, Chunxi Liu, Kjell Schubert, Julian Chan, Pradyot Prakash, Jun Liu, Ching-Feng Yeh, Fuchun Peng, Yatharth Saraf, et al., "Benchmarking LF-MMI, CTC and RNN-T criteria for streaming ASR," in *Proc. SLT*, 2021, pp. 46–51.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[23] Ching-Feng Yeh, Jay Mahadeokar, et al., "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[24] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *Proc. ICASSP*. IEEE, 2020, pp. 7829–7833.

[25] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *Proc. ICASSP*, 2021.

[26] 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. ICASSP*, 2013, pp. 7304–7308.

[27] 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," *arXiv preprint arXiv:2007.03001*, 2020.

[28] Bo Li, Ruoming Pang, Tara N Sainath, Anmol Gulati, Yu Zhang, James Qin, Parisa Haghani, W Ronny Huang, and Min Ma, "Scaling end-to-end models for large-scale multilingual asr," *arXiv e-prints*, pp. arXiv–2104, 2021.

[29] Austin Waters, Neeraj Gaur, Parisa Haghani, Pedro Moreno, and Zhongdi Qu, "Leveraging language id in multilingual end-to-end speech recognition," in *Proc. ASRU*. IEEE, 2019, pp. 928–935.

[30] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.