Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Meta-learning for Indian languages: Performance analysis and improvements with linguistic similarity measures

ANOOP C. S.[1], (Student Member, IEEE), A. G. RAMAKRISHNAN[2], (Senior Member, IEEE)
[1]Indian Institute of Science, Bengaluru, KA 560012 India (e-mail: anoopcs@iisc.ac.in)
[2]Indian Institute of Science, Bengaluru, KA 560012 India (e-mail: agr@iisc.ac.in)

Corresponding author: Anoop C. S.(e-mail: anoopcs@iisc.ac.in).

ABSTRACT Indian languages share a lot of overlap in acoustic and linguistic content. Though different languages use different writing systems, the phoneme sets logically overlap. Most of these languages are low-resourced, lacking enough annotated speech data to build good automatic speech recognition (ASR) systems. Recently proposed model-agnostic meta-learning (MAML) algorithm has shown great success in the fast adaptation of multilingual models to unseen datasets. In this work, we establish the usefulness of MAML pretraining in quickly building reasonably good ASRs for low-resource Indian languages. MAML significantly outperforms joint multilingual training in its capability for few-shot learning and faster adaptation. On average, MAML yields absolute improvements of 5.4% in CER and 20.3% in WER over joint multilingual pretraining in the fast-adaptation setting with five epoch fine-tuning. Further, we exploit the similarities of the source transcriptions with target data through a loss-weighing scheme during the training to improve the performance of MAML models. Similarity-based loss-weighings yield absolute improvements of 0.2% in CER and 1% in WER on average.

INDEX TERMS meta-learning, MAML, language similarity, low-resource ASR, Indian languages, few-shot learning.

I. INTRODUCTION

INDIA is a linguistically diverse country with more than 19,500 languages or dialects as mother tongues. As per the 2011 census data, there are 121 languages in India with more than 10,000 speakers [1]. However, the total numbers of speakers of different languages vary greatly. There are only 14 languages with more than 10 million speakers and only 32 languages with more than 1 million speakers. Generally, the data available to build speech technologies is directly proportional to the number of speakers in that language. Hence, the advancements in speech technologies are not accessible to most Indian languages. Multilingual speech recognition models address this issue by allowing efficient data and parameter sharing across languages.

Conventional ASR systems use three modules: acoustic, pronunciation, and language models, each optimized separately with different objective functions. Hybrid systems [2]–[4] use shared hidden layers to build multilingual acoustic models. End-to-end (E2E) systems eliminate incoherences in optimizations with a single objective function. They allow data pooling from multiple languages, simplifying the multilingual training process [5]–[7]. Multilingual modeling improves the performance of ASR in low-resource settings, and the same model works for all the languages used in training. Conditioning the model on language identity improves the ASR performance significantly but requires additional language information [5], [7]–[9]. The dependency on the prior knowledge of language identity (LID) can be removed by estimating LID jointly in E2E multilingual models [6].

There are other approaches to handling data scarcity in ASR training. They include spectral augmentation [10], adversarial training [11], data synthesis [12], noise addition [13], vocal tract length perturbation [14], [15], and speech and tempo perturbation [16].

Meta-learning has become popular recently. It helps in fast adaptation to unseen data. Model-agnostic meta-learning (MAML) [17] makes the scheme more generic and allows its application to any learning problem trained with gradient descent procedure. MAML was applied to low-resource speech recognition in [18]–[20]. It outperforms joint multilingual pretraining schemes in low-resource settings.

Most Indian languages are low-resourced. Finding enough

native speakers to record the speech data is often difficult. Ensuring the diversity of the data is another challenge. Hence, collecting enough training data is highly time-consuming and expensive for most Indian languages. The writing systems of different languages have non-overlapping character sets. However, there are some opportunities too. There is a significant overlap in the phoneme space of Indian languages. Moreover, unlike some European languages, there is a one-to-one correspondence between the character sets and pronunciation. The languages have slight overlap in their vocabularies too. The value MAML can bring to the ASR performance in low-resource Indian languages is unclear in this context. Also, exploiting the linguistic similarities of the source datasets to the target languages might help improve the performance on the target languages. Our main contributions in this work are as follows:

1) We establish the usefulness of MAML training with a set of pronunciation-based shared labels across multiple Indian languages. The shared labels allow the same pretrained model to be finetuned to many low-resource languages without architectural modifications like changing the output layers.
2) We validate the fast adaptation and few-shot learning capabilities of MAML with a diverse set of 5 target languages chosen from different parts of India, where the pretraining is with four source languages.
3) We compare the performance of MAML in low-resource settings with joint multilingual pretraining and monolingual training.
4) We propose a loss-weighing scheme based on the similarity of source transcriptions with the sentences in the target language to improve the performance of MAML pretraining.

II. RELATED WORK

Meta-learning or *learning to learn* schemes train the model on various learning tasks, such that it can learn any new task faster with only a few training samples. The earlier meta-learning algorithms were based on learning an update function or learning rule [21]–[24], but used additional learnable parameters in the training process. Also, many of them had restrictions on the model architecture [25], [26]. Model-agnostic meta-learning (MAML) proposed in [17] made the learning algorithm more generic. It does not make any assumptions about the model architecture and can easily be applied to any learning problem trained with a gradient descent procedure. It does not introduce additional learnable parameters. We can also use it with various loss functions.

MAML was successful in problems like few-shot classification/regression and meta-reinforcement learning [17]. It finds wide applications in few-shot learning like human pose estimation [27], cancer detection [28], and medical visual question answering [29]. The meta-learner seeks to find an initialization helpful in adapting to various problems. They also adapt quickly (in a few steps) and efficiently (using only a small number of training samples). The training approach in MAML maximizes the sensitivity of the loss functions of the new tasks to the model parameters so that small changes in the parameters yield considerable improvements in the task loss. MAML was applied to low-resource ASR in [18]. They formulated the ASR problem for different languages as different tasks. MAML finds good initializations for ASR models that can easily be fine-tuned to unseen target languages. Authors of [19] suggest a multi-step loss for improving the inner loop optimization of MAML. In [30], authors successfully employ MAML for cross-accented English speech recognition.

There were several attempts to build ASR systems for Indian languages before. The BUT system [31] uses multilingual time-delay neural networks with transfer learning and shows that they perform better than bi-directional residual memory networks (BRMN) and bi-directional LSTM. Their multilingual models have been trained by pooling the data from three languages - Tamil, Telugu, and Gujarati using the common IPA phone set. [32] uses a different strategy for transfer learning. They pretrain the network with large amounts of speech in the high-resource language but with the text transliterated to the target low-resource language. The scheme performs better than transliterating the low-resource data to Latin during fine-tuning. They use the transformer and wav2vec2.0 architectures with English as the high-resource language and six languages, namely Hindi, Telugu, Gujarati, Bengali, Korean, and Amharic, as low-resource targets. Multilingual and code-switching ASR challenge [33] reports an average WER of 32.73% on six Indian languages using TDNN models. [5] uses language conditioning in a listen, attend, and spell (LAS) model jointly trained with around 1500 hours of data from 9 Indian languages and reports an average WER of 21.32%. [9] studies language conditioning in transformer networks and reports the best results with learned language embedding added to the acoustic vectors.

III. MAML FOR LOW-RESOURCE SPEECH RECOGNITION

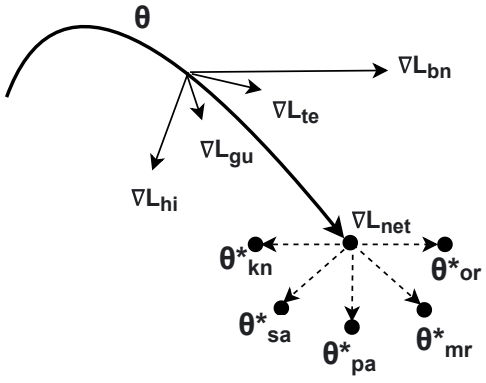


FIGURE 1: Model-agnostic meta-learning (MAML) employed for fast adaptation of ASR models to low-resource languages. *hi*, *gu*, *te*, and *bn* denote the source languages of Hindi, Gujarati, Telugu, and Bengali, respectively. *kn*, *sa*, *pa*, *mr*, and *or* denote the target languages of Kannada, Sanskrit, Punjabi, Marathi, and Odia.

Consider an ASR model f_θ parameterised by θ. We need θ that can easily be fine-tuned to any new language. We have a few datasets for the high resource languages $\mathcal{D} = \{D_1, D_2, \ldots, D_L\}$, where L denotes the number of high resource languages used for meta-training. Every model update cycle consists of several inner optimizations and one round of outer optimization. During inner optimizations, we iterate over all the datasets D_l, $l = 1, 2, \ldots, L$ and the number of optimizations equals the number of datasets L in general. We sample a batch of examples B_l from its train set and divide it into two sets, B_l^{tr} and B_l^{dev}, with an approximately equal number of utterances. The model parameters θ are updated to θ' by computing the loss function on B_l^{tr} and performing one gradient descent step.

$$\theta_l' = \theta - \alpha \nabla_\theta \mathcal{L}_{B_l^{tr}}(f_\theta) \tag{1}$$

where α is the learning rate of the inner optimizer. The loss with respect to $f_{\theta'}$ is computed on the unseen part of the data B_l^{dev}, and the gradients are computed based on that. This process is repeated for all the training languages, and the accumulated gradients are used to update the original weights f_θ during the outer optimization. Thus the objective of meta-training is to optimize the performance of the fine-tuned model $f_{\theta_l'}$ on unseen data B_l^{dev} and can be summarised as follows:

$$\min_\theta \sum_{B_l} \mathcal{L}_{B_l^{dev}}(f_{\theta_l'}) = \sum_{B_l} \mathcal{L}_{B_l^{dev}}(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{B_l^{tr}}(f_\theta)}) \tag{2}$$

The outer optimization is performed as follows:

$$\theta = \theta - \beta \sum_{B_l} \nabla_\theta \mathcal{L}_{B_l^{dev}}(f_{\theta_l'}) \tag{3}$$

where β is the step size for the outer optimizer and $f_{\theta_l'}$ is the network adapted on language l. Using first-order approximation of MAML, the outer optimization is reformulated as:

$$\theta = \theta - \beta \sum_{B_l} \nabla_{\theta_l'} \mathcal{L}_{B_l^{dev}}(f_{\theta_l'}) \tag{4}$$

The whole algorithm can be summarised as follows:

Algorithm 1 Model-agnostic meta-learning

REQUIRE: α, β: Step-size hyperparameters
1: Randomly initialize θ
2: **while** not done **do**
3: **for each** $D_l \in \mathcal{D}$ **do**
4: Sample a batch of data B_l from D_l
5: Split B_l to B_l^{tr} and B_l^{dev}
6: Evaluate $\nabla_\theta \mathcal{L}_{B_l^{tr}}(f_\theta)$
7: Compute adapted parameters with gradient descent: $\theta_l' = \theta - \alpha \nabla_\theta \mathcal{L}_{B_l^{tr}}(f_\theta)$
8: **end for**
9: Update $\theta \leftarrow \theta - \beta \sum_{B_l} \nabla_{\theta_l'} \mathcal{L}_{B_l^{dev}}(f_{\theta_l'})$
10: **end while**

We use $L = 4$ with Hindi (*hi*), Gujarati (*gu*), Telugu (*te*), and Bengali (*bn*) as the source languages. Fig. 1 illustrates the gradient update procedure employed in our work.

IV. EXPERIMENTAL SETUP

A. SPEECH DATASETS USED FOR THIS WORK

We use Hindi (*hi*), Gujarati (*gu*), Telugu (*te*), and Bengali (*bn*) as the source languages for pretraining the joint multilingual and MAML models. The languages are chosen such that there is sufficient representation from the northern (*hi*), western (*gu*), southern (*te*), and eastern (*bn*) parts of India. The choice is also based on the availability of off-the-shelf datasets in these languages. As per 2011 census data, these languages cover around 79% of the Indian population. We use Marathi (*mr*), Punjabi (*pa*), Odia (*or*), Sanskrit (*sa*), and Kannada (*kn*) as the target languages. The details of the datasets are given in Table 1. All the datasets are publicly available and can be downloaded from the links provided in the citations. The validation subsets of these datasets are used only for assessing the performance of the models across the training epochs.

B. ENCODING THE TEXT DATA IN SLP1 FORMAT

Most Indian languages share a nearly common phoneme space. But most of them have their own writing systems with non-overlapping character sets. For example, the word *guru* is pronounced almost similarly across Indian languages, but the text representations are completely different as shown in Fig. 2.

Latin	Hindi	Bengali	Telugu	Gujarati
Guru	गुरु	গুরু	గురు	ગુરુ

FIGURE 2: Representation of the word *Guru* in four Indic scripts.

The standard method of E2E multilingual training is to pool the character tokens from multiple languages [5], [6]. However, this leads to inefficient data sharing while training the acoustic model, since identical phoneme sequences are represented with different characters in different languages.

Common, language-agnostic tokens have been proposed in [8], [41], [42] for efficient data-sharing across related languages during multilingual training. Authors of [41] propose a language-agnostic multilingual ASR system that transforms all languages into one writing system through a many-to-one transliteration transducer. In [42], the authors model the text as a sequence of Unicode bytes. Using bytes helps keep the output dimensions fixed (vocabulary size is always limited to 256 in this case), thereby eliminating the need for changes to the output layer with the incorporation of new languages.

In Indian languages, there is mostly a one-to-one association between the character sets and their pronunciations. Also, the Unicode tables for Indian languages are structured such that the grapheme units with similar pronunciations

TABLE 1: Details of the speech datasets used for building acoustic models. The links to the sources of the datasets are provided in the references.

			# Utterances			Duration (hrs)		
	Language	**Dataset**	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
Source	Hindi (*hi*)	SLR118 [34]	36308	1744	-	100	5	-
	Bengali (*bn*)	Common Voice 10.0 [35]	29273	3000	-	50	5	-
	Telugu (*te*)	MUCS2021 [33]	47418	3037	-	45	5	-
	Gujarati (*gu*)	MUCS2021 [33]	26067	3075	-	45	5	-
Target	Marathi (*mr*)	Common Voice 10.0 [35]	5734	600	600	10	1	1
	Punjabi (*pa*)	Shrutilipi (AI4Bharat) [36]	5560	522	528	12.5	1	1
	Odia (*or*)	SME [37]	5581	1218	719	7	1.5	1
	Sanskrit (*sa*)	Vāksañcayaḥ [38]	6000	500	500	10	1	1
	Kannada (*kn*)	SLR126 [39], [40]	3000	500	500	10	1.5	1.5

Language	Unicode range / Offset ->	02	03	05	06	07	08	15	19	2F	30	3E	3F
Hindi	U+0900 – U+097F	◌ं	◌ः	अ	आ	इ	ई	क	ङ	य	र	◌ा	◌ि
Bengali	U+0980 – U+09FF	◌ং	◌ঃ	অ	আ	ই	ঈ	ক	ঙ	ষ	র	◌া	◌ি
Telugu	U+0C00 – U+0C7F	◌ం	◌ః	అ	ఆ	ఇ	ఈ	క	ఙ	య	ర	◌ా	◌ి
Gujarati	U+0A80 – U+0AFF	◌ં	◌ઃ	અ	આ	ઇ	ઈ	ક	ઙ	ય	ર	◌ા	◌િ
SLP1		M	H	a	A	i	I	ka	Na	ya	ra	A	i

FIGURE 3: Using shared labels across multiple Indian languages with SLP1 transliteration scheme. Graphemes (characters) of different languages representing the same phoneme are at the same offset in their assigned Unicode space and share the same SLP1 codes.

from different languages occur at the same offset from the beginning of the respective tables. Common label sets (CLS) [43] and Sanskrit library phonetic encoding (SLP1) [44] are two popular schemes for generating language-independent tokens for multilingual ASR training in Indian languages. Both exploit the structure of the Unicode tables. In [45] and [46], the authors represent similar phonetic units through the common label set. SLP1 tokens are used for monolingual ASR training in [38], [47] and multilingual training in [48]. Examples of the SLP1 transliteration scheme for our source languages are shown in Fig. 3.

SLP1 encoding can be inferred from the Unicode for the character based on its offset from the beginning of the range assigned to that language. It maps vowels and vowel modifiers (symbols used when vowels are not at the beginning of the words) to the same tokens, eliminating redundancy in the native characters. Due to its simplicity in implementation, we use SLP1 in this work. The complete mapping of native characters to SLP1 for the languages used in our study is shown in Appendix A. The SLP1 transliteration scheme yields 54 tokens representing the characters from all the Indian languages covered in our work.

Some special characters like anusvara, visarga, and nukta, and properties like schwa deletion cause slight modifications to the pronunciation of words in Indian languages. We handle such deviations as suggested in [49] and [50] to ensure maximum correlation between the token sequences and the underlying acoustic data.

C. ARCHITECTURE FOR MULTILINGUAL TRAINING AND MAML

We use an end-to-end (E2E) architecture for joint multilingual training and meta-learning, as shown in Fig. 4. Similar to [51], we use an upstream-downstream setup. We use IndicWav2Vec [52] large model in the upstream. It acts as a front-end feature extractor. IndicWav2Vec-large is pretrained on 17,000 hours of raw speech data from 40 Indian languages. The 1024-dimensional output of the model is converted to 80 dimensions by a linear layer and fed to the downstream ASR task. The parameters of the upstream model are fixed during the training phase of the ASR.

Conformer [53] encoders and transformer [54] decoders are used for acoustic modeling in the downstream ASR. The hyperparameters of the model are listed in Table 2. Joint connectionist temporal classification (CTC) [55] - attention scheme [56] is used for training the model with a CTC weight of 0.3 and an attention weight of 0.7. We use Adam optimizer [57] with a peak learning rate of 0.005 and warm-up steps of 30000 during the monolingual and joint multilingual training. For MAML training, we use stochastic gradient descent (SGD) with a learning rate of 0.0001 for inner loop optimization. Adam is used for outer loop optimization. We train the model for 25 epochs in the pretraining phase. The resulting model is fine-tuned for 5 epochs using the data from the target languages. No language models are used in our experiments.

We use beam-search decoding and compute the most prob-

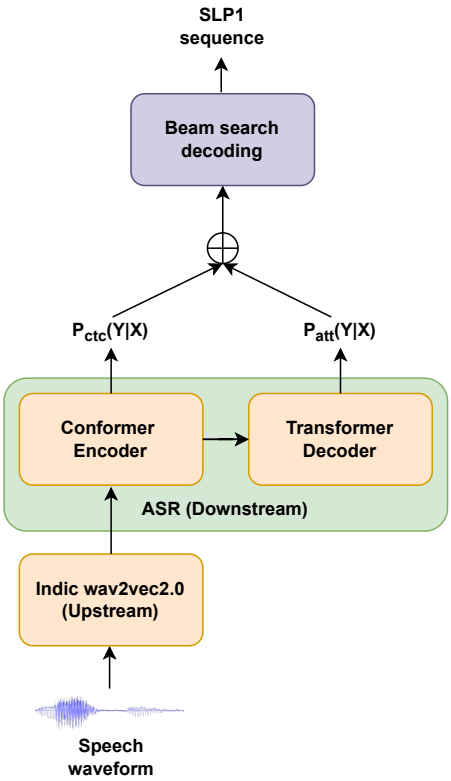


FIGURE 4: End-to-end (E2E) architecture employed for joint multilingual training and MAML in our experiments.

TABLE 2: Hyperparameters of the conformer used in acoustic modeling.

Parameters	Value
Kernel size	15
Encoder layers	12
Decoder layers	6
Attention heads	4
Attention dimension	256
Feed-forward dimension	2048

able output sequence $\hat{Y}$ as:

$$\hat{Y} = \underset{Y \in V^*}{\arg\max} \Big(\alpha \, log \, p_{ctc}(Y|X) + (1-\alpha) log \, p_{att}(Y|X) \Big) \quad (5)$$

where p_{ctc} and p_{att} are the CTC and attention scores, α is the CTC weight and V^* is the set of all possible target hypotheses. We use a beam size of 20 and α = 0.3 during decoding. All the experiments are conducted using the ESPNet toolkit [58].

V. RESULTS AND ANALYSIS

In this section, we discuss our results on the fast adaptation and few-shot learning capability of MAML in the context of Indian languages. Further, we investigate the effects of varying the number of epochs in the pretraining and fine-tuning phases.

A. PERFORMANCE OF MAML IN FAST-ADAPTATION SETTINGS

We use three different initializations for fine-tuning the ASR model to low-resource languages.

1) Random (no pretraining) initialization is used, and the monolingual models are trained from scratch.
2) A joint multilingual model pretrained for 25 epochs is used as the initial model. The data from all the source languages are employed for training using the standard ESPNet [58] recipes.
3) A meta-learning model pretrained for 25 epochs is used as the initialization.

The initial model is fine-tuned for five epochs using the limited data available for the target languages. We compare the performances of these models in Table 3.

The random initialization (monolingual training) yields the worst performance and seems incapable of learning good model weights with a few epochs (5 in this case). With around 10 hours of speech data in target languages and just five epochs of training, monolingual models fail to perform reasonably for any language. Both joint multilingual and MAML pretraining schemes provide significant performance improvements. It can be seen from Table 3 that MAML provides the best initialization for the model weights. Its performance is significantly better than that of joint multilingual training. On average, there are absolute improvements of 5.4% in CER and 20.3% in WER. The maximum benefits are seen for Odia, where the training data is extremely limited (just 7 hours). The results suggest that pretraining with meta-learning is extremely useful for fast adaptation to multiple languages when the target languages are closely related to source languages, as in the case of languages of the Indian subcontinent.

B. EFFECT OF THE SIZE OF TRAINING DATA ON THE PERFORMANCE OF TARGET LANGUAGE ASR

Here we study the few-shot capability of our joint multilingual and MAML pretrained models. We choose 0%, 25%, 50%, and 100% of the target data as subsets. Since we use SLP1 tokens in the output layer for all languages, the models pretrained on source languages can be used directly for target languages without finetuning. We refer to this case as *0% shot*. In other cases, we perform finetuning for five epochs with the subsets and evaluate the performance of the resulting ASR. The zero-shot case is expected to fare poorly compared to the rest. The results are shown in Table 4.

The average zero-shot performance of the models pretrained with MAML is almost similar to joint multilingual pretraining. The average zero-shot CER for MAML and joint training are 27.4% and 28.5%, respectively. The CER of the MAML pretrained model becomes almost half that of joint multilingual pretrained model after fine-tuning with just 25% (2 to 3 hrs) of the target dataset. The absolute improvements in CER (WER) are 12.0% (34.5%), 17.1% (40.4%), 17.9% (35.0%), 13.9% (41.4%), and 13.8% (53.2%) for Marathi,

TABLE 3: CER and WER for ASR models trained/fine-tuned for five epochs using low-resourced target datasets.

Model	CER (%)					WER (%)				
	mr	pa	or	sa	kn	mr	pa	or	sa	kn
Random initialization (no pretraining)	82.5	140.9	98.2	84.8	98.3	156.8	114.1	123.0	106.7	100.0
Joint multilingual pretraining	13.0	14.2	17.1	8.3	9.2	53.6	40.7	62.7	56.0	53.8
MAML pretraining	**8.3**	**6.3**	**8.4**	**6.1**	**5.5**	**35.6**	**17.5**	**34.2**	**43.2**	**34.9**
Absolute improvements with MAML over joint multilingual pretraining	4.7	7.9	8.7	2.2	3.7	18.0	23.2	28.5	12.8	18.9

TABLE 4: CER and WER for ASR models with few-shot fine-tuning for five epochs on target datasets. Performance of MAML pretraining with 25% of the data is comparable to that of joint multilingual training with the complete data.

Target Language	MAML Pretraining				Joint Multilingual Pretraining			
	0%-shot	25%-shot	50%-shot	100%-shot	0%-shot	25%-shot	50%-shot	100%-shot
CER (%)								
mr	25.6	13.6	10.3	8.3	31.5	27.1	16.0	13.0
pa	31.0	13.9	9.0	6.3	29.5	26.5	20.3	14.2
or	34.6	16.7	10.8	8.4	39.7	34.3	24.2	17.1
sa	22.8	8.9	7.0	6.1	22.1	17.0	10.5	8.3
kn	22.8	9.0	7.0	5.5	19.6	17.2	11.3	9.2
WER (%)								
mr	90.3	55.8	44.0	35.6	93.6	86.7	63.6	53.6
pa	82.7	42.3	26.7	17.5	76.1	71.5	59.0	40.7
or	99.6	64.6	43.7	34.2	109.2	100.0	80.2	62.7
sa	99.5	58.1	48.4	43.2	102.8	84.3	63.9	56.0
kn	107.1	53.9	44.1	34.9	92.0	82.8	63.2	53.8

Punjabi, Odia, Sanskrit, and Kannada, respectively, in the case of MAML pretraining for 25%-shot fine-tuning. The corresponding absolute improvements in joint multilingual pretraining are 4.4% (6.9%), 3.0% (4.6%), 5.4% (9.2%), 5.1% (18.5%), and 2.4% (9.2%), respectively. The average absolute improvements in CER (WER) are 14.9% (40.9%) for MAML pretraining and 4.1% (9.7%) for joint multilingual pretraining. It is also interesting to note that the average CER performance of 25% shot MAML (12.4%) closely matches that of 100%-shot joint training, and the WER is off-target by just 1.6%. These figures illustrate the benefits of MAML over joint multilingual pretraining in the case of few-shot fine-tuning for low-resource languages.

Comparing the 0%-shot and 100%-shot columns, we can see that the average absolute improvements in CER (WER) with just five epochs of fine-tuning are 20.4% (62.8%) for MAML and 16.1% (41.4%) for joint multilingual pretraining. These values support the fast adaptation property of MAML to any new dataset.

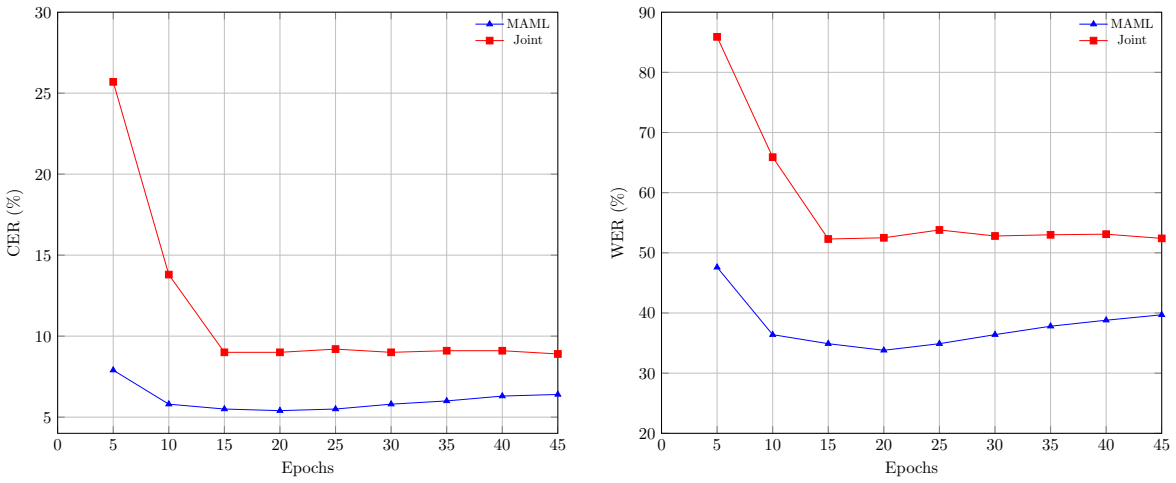
C. EFFECT OF THE NUMBER OF PRETRAINING EPOCHS

In this section, we study the effect of the number of pretraining epochs in MAML and joint pretraining on the ASR performance on low-resourced languages. Models are trained for 45 epochs using the speech data from source datasets. Intermediate models are saved every five epochs. They are fine-tuned for five epochs using the data from target languages, and the performances of the resulting models are evaluated in terms of CER and WER. The results are shown for Kannada in Fig. 5. It can be seen that only around 15 to 25 pretraining epochs are necessary for reasonable fine-tuning to target languages. After that, the fine-tuning performance degrades. Since the behavior is similar in other target languages, we use 25 epochs of pretraining in all our experiments.

D. EFFECT OF THE NUMBER OF FINE-TUNING STEPS

In this section, we study the effect of fine-tuning steps on the performance of the ASR model. We use three initial models as in Section V-A. We fine-tune the initial model for 5, 10, 15, and 20 epochs and evaluate the performance of the resulting ASR. The results are shown in Figs. 6 and 7. For MAML and joint multilingual pretrained models, we also show the CER and WER values with the initial model (0 epoch). The errors for MAML and joint pretraining are always lower than those for monolingual training showing the significance of pretraining in low-resource speech recognition. MAML error rates are lower than those for joint multilingual pretraining for every number of pretraining epochs (5, 10, 15, and 20) considered, which shows the fast-fine-tuning capability of MAML. MAML curves also move towards saturation faster.

Table 5 summarises the performance of these models after fine-tuning for 20 epochs. The joint pretraining fails to reach the performance of 5-epoch fine-tuned MAML models for most languages, even after 20 epochs. Comparing 20-epoch fine-tuned models, MAML beats the joint pretraining by



(a) (b)

FIGURE 5: Effect of the number of pretraining steps on the performance of Kannada ASR: a) CER and b) WER. Blue triangles: MAML pretraining; red squares: joint multilingual pretraining.

TABLE 5: CER and WER for ASR models trained/fine-tuned for 20 epochs using low-resource target datasets. Performance of MAML pretraining is clearly superior to that of joint multilingual pretraining.

Model	CER (%)					WER (%)				
	mr	pa	or	sa	kn	mr	pa	or	sa	kn
Random initialization (no pretraining)	10.7	9.0	12.2	9.8	10.0	44.3	22.8	46.5	58.5	55.4
Joint multilingual pretraining	8.5	7.7	9.5	6.0	5.7	34.7	20.0	34.9	41.8	34.7
MAML pretraining, fine-tuned 20 epochs	5.7	1.6	5.8	5.2	4.4	23.6	3.3	23.1	38.0	28.4
MAML pretraining, fine-tuned 5 epochs	8.3	6.3	8.4	6.1	5.5	35.6	17.5	34.2	43.2	34.9

an absolute 2.9% in CER and 9.9% in WER on average. The average absolute improvements over direct monolingual training are 5.8% in CER and 22.2% in WER.

VI. EXPLOITING LANGUAGE SIMILARITIES IN MAML

There is considerable overlap among the lexical and acoustic content of different sets of Indian languages due to the language family relationships. We study whether such similarities can be exploited to improve the average performance of MAML for a set of target languages. We first investigate the importance of source language selection in MAML. Then we examine whether similarities between sentences in the source and target languages can be exploited for performance improvements in ASR for target languages.

A. IMPORTANCE OF SOURCE LANGUAGE SELECTION

We have used Hindi, Bengali, Telugu, and Gujarati as source languages representing the northern, eastern, southern, and western parts of India, in that order. The target languages are Marathi, Punjabi, Odia, Sanskrit, and Kannada. The native speakers of all our target languages, except Sanskrit, are largely confined to particular states in India. There are only limited speakers of Sanskrit, and they are distributed over different geographic areas. The geographic locations where the source and target languages are spoken are shown in Fig. 8. To understand the importance of the selection of source languages for pretraining, we remove one of the source languages at a time and evaluate the ASR performance in the target languages. The results are shown in Fig. 9. The first bar shows the results when all four languages are included in pretraining. The remaining bars are the results of pretraining with only three source languages, with a different fourth language excluded each time. Based on Fig. 9, we deduce the most impactful pretraining language for each target language and summarise the results in Table 6.

TABLE 6: Most impactful source language for the performance of ASR in each target language and the degree of impact (absolute degradation or increase in CER or WER) when that language is removed from the pretraining set.

Target language	Most impactful Source language	Degradations in	
		CER	WER
Marathi	Gujarati	0.5	2.1
Punjabi	Hindi	0.5	1.7
Odia	Bengali	0.9	3.6
Sanskrit	Telugu	0.1	1.3
Kannada	Telugu	0.9	4.5

Table 6 indicates that the impact of the removal of a source

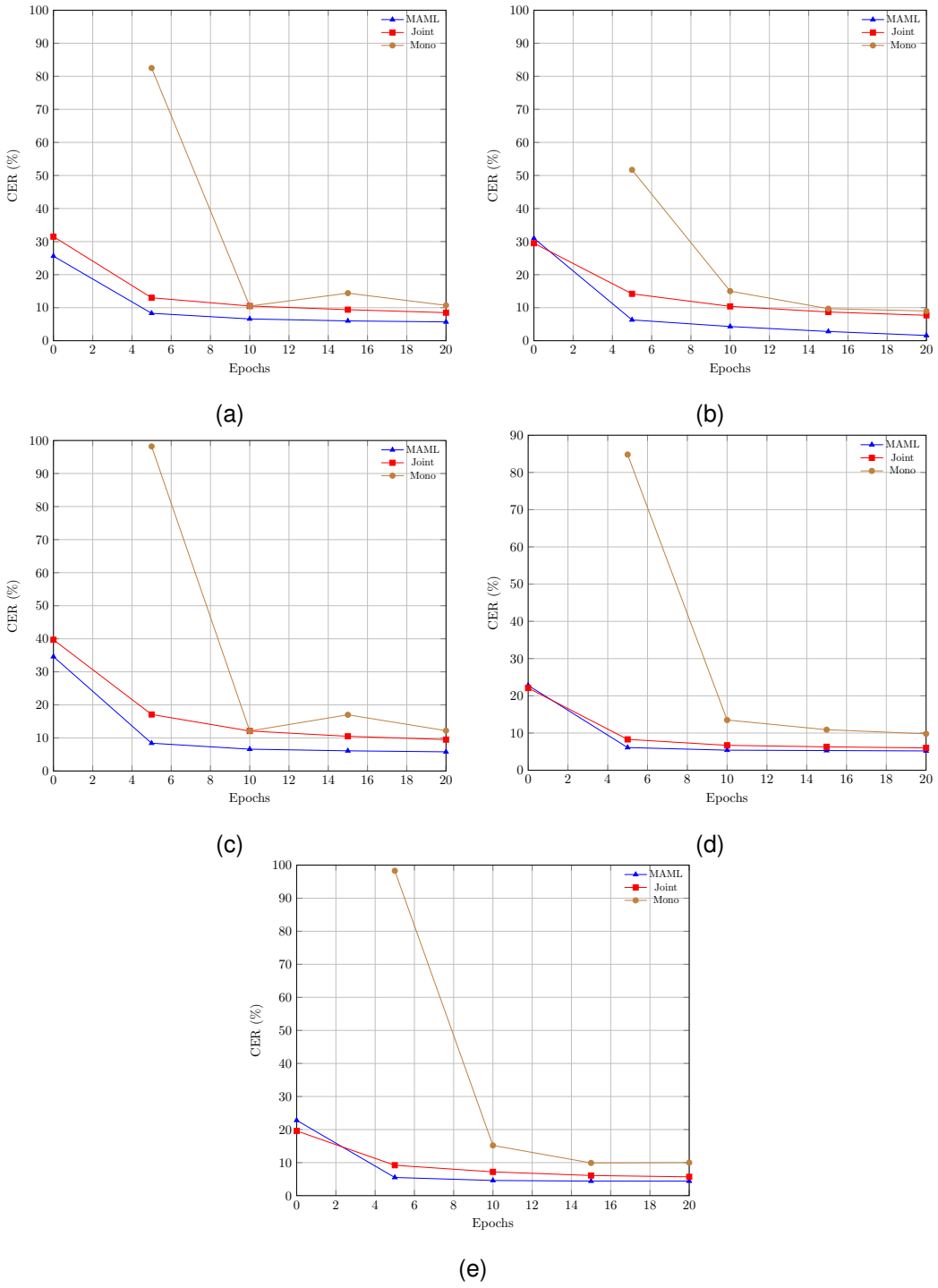


FIGURE 6: CER (in %) Vs. the number of fine-tuning epochs (0, 5, 10, 15, 20) for the different target languages: (a) Marathi (b) Punjabi (c) Odia (d) Sanskrit (e) Kannada. Blue triangles: MAML pretraining; Red squares: joint multilingual pretraining; Brown circles: monolingual training.

language on a target language ASR depends on the geographic closeness of the regions where they are spoken. For example, the performance of Odia ASR degrades the most when Bengali is removed from the pretraining languages. Similarly, the performance of Kannada ASR degrades the most when Telugu is removed from the pretraining languages. For Marathi and Punjabi, the maximal degradations happen with the removals of Gujarati and Hindi, respectively. Fig. 9 gives some other useful information. Sanskrit is not impacted much by the removal of any of the source languages. This can easily be correlated to the fact that the speakers of Sanskrit are not confined to any specific geographic lo-

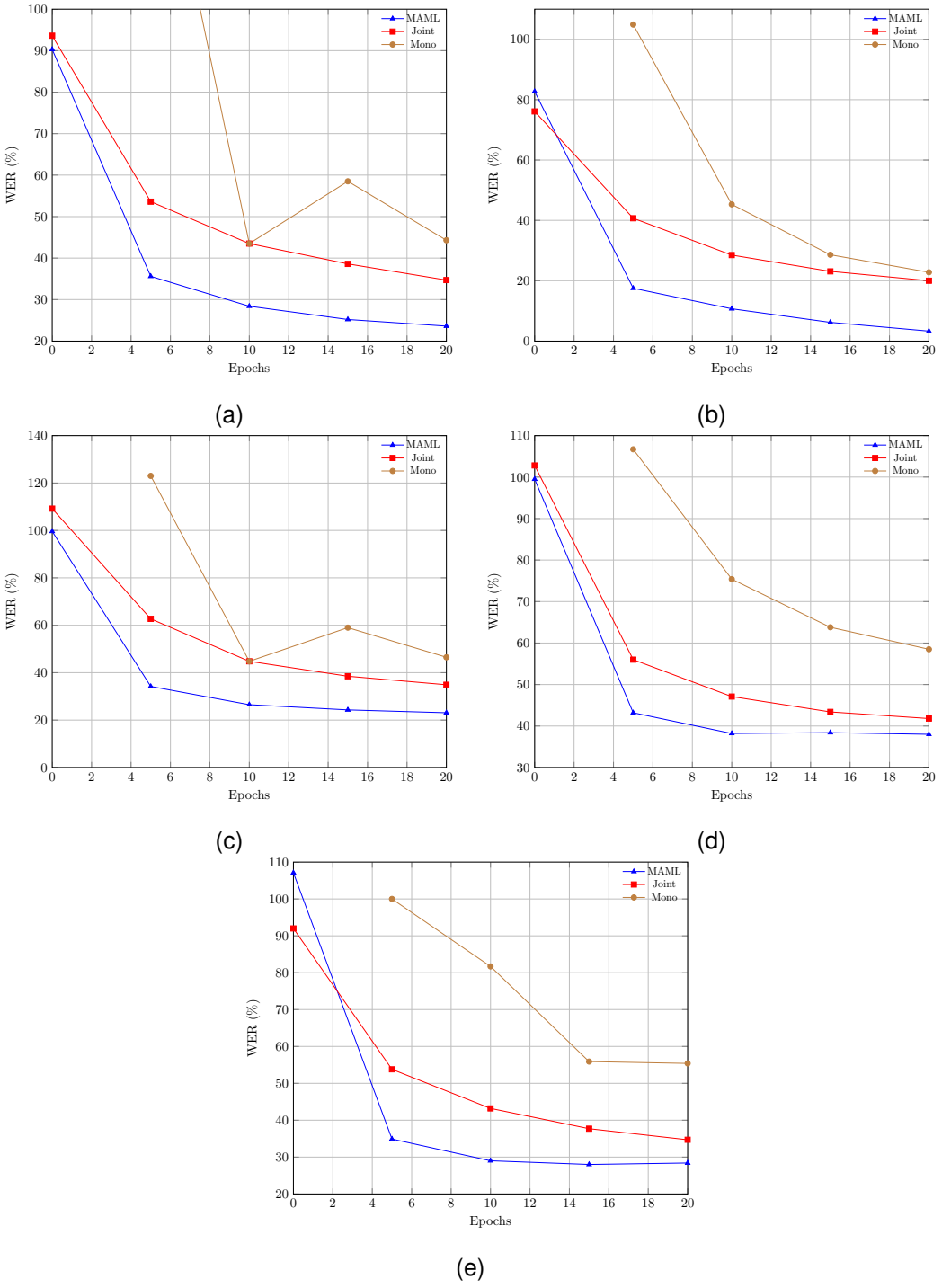


FIGURE 7: WER (in %) Vs. the number of fine-tuning epochs (0, 5, 10, 15, 20) for the different target languages: (a) Marathi (b) Punjabi (c) Odia (d) Sanskrit (e) Kannada. Legends same as that of Fig. 6.

cation in India. When Gujarati is removed from the source languages, the performance of Sanskrit ASR improves. This indicates that Gujarati may have minimal acoustic similarity with Sanskrit. Kannada has no implicit schwa deletion, and the performance degradations due to the removal of schwa deletion-based languages - Hindi, Bengali, and Gujarati- are almost similar.

B. CAPTURING THE TEXT SIMILARITIES IN THE SLP1 SPACE

In section VI-A, we have seen that choosing a pretraining language close to the geographic location where the target language is spoken helps improve the MAML performance on the target language. This motivates us to use the similarity between the source and target languages to improve the MAML pretraining. For this, we need an embedding of the

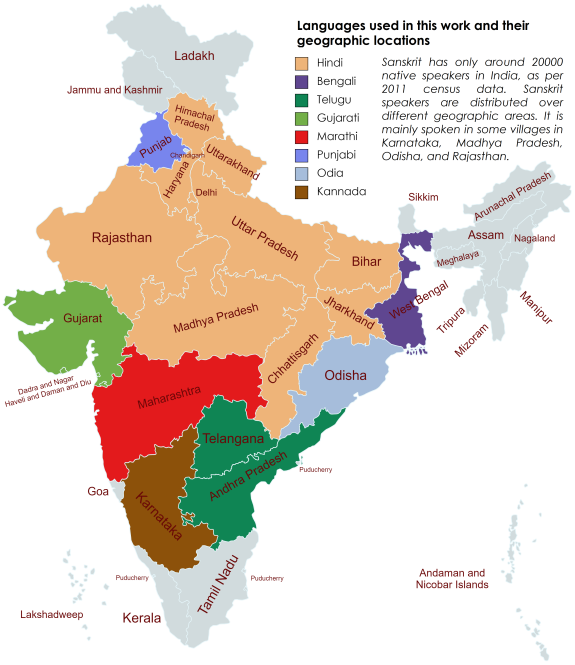


FIGURE 8: Languages used in this work and the geographies where they are spoken. This map is taken from the Geographical Society of India.

source and target sentences in the SLP1 space, which can reliably capture such similarities.

We use fastText [59] to build the word embeddings in the SLP1 space. The publicly available AI4Bharat IndicCorp [60] is used as the text corpus to train fastText models. The size of the text corpus in each language is listed in Table 7. There are around 164 million utterances in the corpus. We use the skip-gram model for training fastText. For sentence embeddings, we use the mean vector of word embeddings. To evaluate the suitability of fastText embeddings for our application, we compute the average cosine similarities and Euclidean distances between the sentences in our training sets. The cosine similarity between two sentence vectors $\mathbf{u}$ and $\mathbf{v}$ is computed as:

TABLE 7: Details of AI4Bharat IndicCorp [60] text corpus used for building fastText embeddings

Language	**# Utterances** (in million)
Hindi (*hi*)	32.4
Bengali (*bn*)	20.3
Telugu (*te*)	26.1
Gujarati (*gu*)	20.9
Marathi (*mr*)	19.0
Punjabi (*pa*)	12.4
Odia (*or*)	4.3
Sanskrit (*sa*)	0.4
Kannada (*kn*)	28.0

$$Cosine\ similarity(\mathbf{u},\mathbf{v}) = \frac{1}{2}\left(1 + \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\,\|\mathbf{v}\|}\right) \quad (6)$$

The results are shown in Table 8(a). The order of similarity of source datasets with the target languages based on Table 8(a) are shown in Table 8(b). The data in Table 8(b) reflects the geographies in which the target languages are spoken. To further establish the usefulness of fastText embeddings in capturing the similarities between sentences from different languages, we plot the two-dimensional UMAP (uniform manifold approximation and projection) [61] of transcriptions in the source and target datasets. UMAP is known for its capability to find the critical structures in the high-dimensional space and preserve them in the low-dimensional space. 2-D UMAP plot of fastText embeddings of sentences in the pretraining and target datasets is shown in Fig. 10. The black dot at the center of the clusters in the figure denotes the centroid of the sentence embeddings in that dataset. The Marathi, Punjabi, Odia, Sanskrit, and Kannada clusters are near the Gujarati, Hindi, Bengali, Hindi, and Telugu clusters, respectively. Also, Sanskrit is closer to Hindi and Telugu than the other source languages. The observations reflect the relative positions of the geographic regions to which the native speakers of these languages belong [62]. Table 8 and Fig. 10 suggest that the fastText embeddings can capture the similarity between sentences in the SLP1 space.

C. EXPLOITING THE TEXT SIMILARITIES IN MAML PRETRAINING

In [63], [64], language level similarities were explored for data selection to derive multilingual bottleneck features in hybrid ASR systems. Authors of [65] propose a data-weighing strategy on the training samples based on the posterior of the target language extracted from a language classifier. All these works are intended to improve multilingual speech recognition in a specific target language. In task similarity aware MAML (TSA-MAML) [66], authors use multiple group initializations instead of a single initialization for all the tasks. The group initializations are based on task similarity estimated from the Euclidean distance between task-specific model parameters. They use it to show improvements in few-shot image classification tasks.

In this work, we propose a weighing strategy based on the similarity of the source transcriptions with the SLP1 sentences from the target languages. Unlike the earlier schemes [65], our weighing strategy aims to improve MAML pretraining to increase the average ASR performance across all the target languages. This approach is motivated by the study in [67], where the authors derive an upper bound for the average absolute meta-generalization gap based on the upper bound for the divergence between data distributions of any two tasks sampled from the task distribution. However, instead of using a data selection approach to ensure the closeness of languages (tasks) as in [64], we use a weighing strategy to give more importance to the source samples "close" to the data distribution of the target languages. We measure "closeness" in terms of Mahalanobis distance and cosine similarity. Unlike [66], where the distances are measured in model parameter space, we make measurements in the SLP1 space.

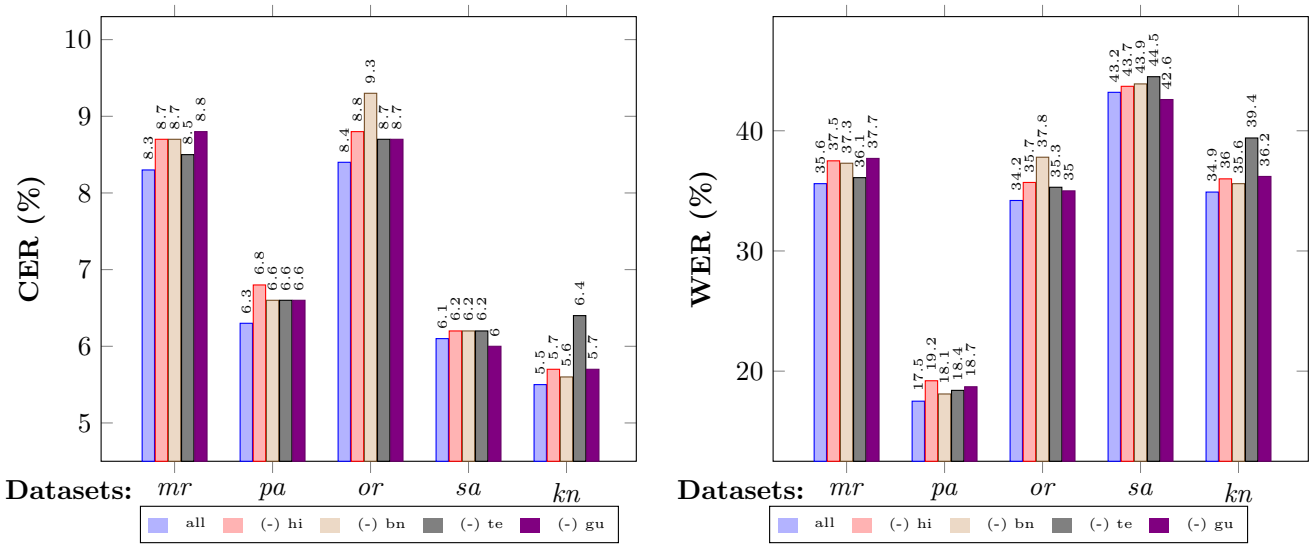


FIGURE 9: Effect of source language selection in meta-learning. a) CER and b) WER (in %) of the ASR in target languages when one source language is excluded from pretraining. hi: Hindi; bn: Bengali; te: Telugu; gu: Gujarati.

TABLE 8: Measures of similarity between the source and target datasets estimated from fastText sentence embeddings

(a) Average cosine similarity and Euclidean distance between each pair of source and target languages. The highest similarity values are marked in bold.

Languages	Cosine similarity				Euclidean distance			
tgt (↓) / src (→)	hi	bn	te	gu	hi	bn	te	gu
Marathi	**0.802**	0.720	0.672	0.777	**0.651**	0.783	0.913	0.703
Punjabi	**0.851**	0.729	0.706	0.777	**0.556**	0.759	0.855	0.693
Odiya	0.739	**0.824**	0.718	0.727	0.788	**0.654**	0.881	0.818
Sanskrit	0.702	0.717	**0.761**	0.694	0.837	0.825	**0.810**	0.863
Kannada	0.674	0.696	**0.761**	0.667	0.836	0.816	**0.780**	0.861

(b) Order of similarity of source languages with the target languages.

Tgt.	Order of similarity
mr	hi > gu > bn > te
pa	hi > gu > bn > te
or	bn > hi > gu > te
sa	te > bn > hi > gu
Kn	te > bn > hi > gu

TABLE 9: CER and WER for MAML-pretrained ASR models exploiting text similarities in the SLP1 space. The best values are highlighted in bold.

Model	CER (%)						WER (%)					
	mr	pa	or	sa	kn	avg.	mr	pa	or	sa	kn	avg.
5-epoch fine-tuned MAML model	8.3	**6.3**	8.4	6.1	5.5	6.9	35.6	**17.5**	34.2	43.2	34.9	33.1
+ Mahalanobis distance based similarity	**8.0**	**6.3**	**8.0**	**5.8**	**5.3**	**6.7**	34.7	17.6	**32.3**	41.8	**33.6**	**32.0**
+ Cosine similarity	8.1	6.4	**8.0**	**5.8**	**5.3**	**6.7**	**34.6**	17.7	32.4	**41.5**	34.1	32.1

1) Mahalanobis distance-based similarity measurements

We build a Gaussian mixture model (GMM) to model the data distribution of all the target languages together. Text corpus from AI4Bharat IndicCorp [60] is used for GMM training. Each sentence from the text corpus is first converted to SLP1 and embedded using fastText models learned in Section VI-B. A 15-component GMM with diagonal covariance matrices is trained using the fastText embeddings. k-means++ is used to initialize the GMM. The learned GMM is of the form:

$$p(\boldsymbol{t}) = \sum_{k=1}^{K} \lambda_k \mathcal{N}(\boldsymbol{t} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{7}$$

where $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and λ_k represent the mean, covariance matrix and mixing coefficient for the k^{th} Gaussian. The number of Gaussians $K = 15$ in our case. $\boldsymbol{t}$ is the fastText embedding for the SLP1 sentence in the target text corpus.

For each source transcription, the Mahalanobis distance is calculated from every Gaussian component in the GMM. If $\boldsymbol{s}^{(i)}$ is the fastText embedding for the i^{th} speech transcription in the source dataset, the Mahalanobis distance $d_k^{(i)}$ from the

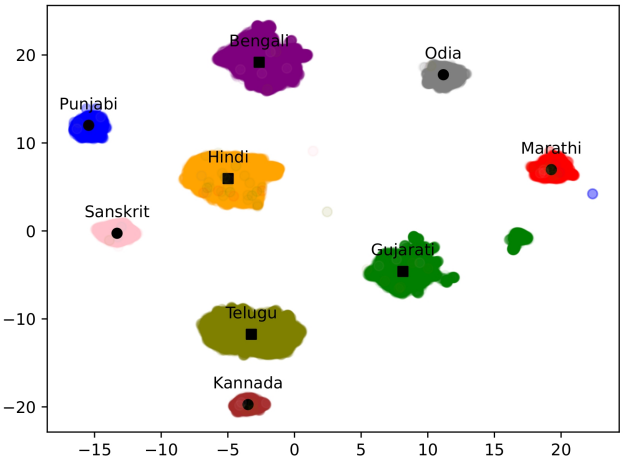


FIGURE 10: UMAP plot showing similarities between the source and the target languages using fastText representation of the text data in the SLP1 space. The black dot at the center of each cluster denotes the centroid of the sentence embeddings in that dataset.

k^{th} Gaussian is computed as:

$$d_k^{(i)} = \sqrt{\left(\boldsymbol{s}^{(i)} - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1} \left(\boldsymbol{s}^{(i)} - \boldsymbol{\mu}_k\right)}, \quad k = 1, 2, \ldots, K \text{ and } i = 1, 2, \ldots, N. \tag{8}$$

where N is the total number of utterances in the source corpus. These distances are weighted by the respective mixing coefficients to arrive at a single distance measure $d^{(i)}$.

$$d^{(i)} = \sum_{k=1}^{K} \lambda_k d_k^{(i)}, i = 1, 2, \ldots, N \tag{9}$$

The calculated distances are first scaled to the [0, 1] range ($d_{norm}^{(i)}$) and then subtracted from 1 to get a similarity measure $v^{(i)}$.

$$d_{norm}^{(i)} = \frac{d^{(i)} - d_{min}}{d_{max} - d_{min}}, \; i = 1, 2, \ldots, N \tag{10}$$

$$\text{where } d_{max} = max\{d^{(i)}, i = 1, 2, \ldots, N\}$$

$$\text{and } d_{min} = min\{d^{(i)}, i = 1, 2, \ldots, N\}.$$

$$v^{(i)} = 1 - d_{norm}^{(i)}, \; i = 1, 2, \ldots, N \tag{11}$$

The resulting values $v^{(i)}$ are then moved to the [-1, 1] range, and sigmoid is applied to get the final weight $w^{(i)}$ for each transcription in the source dataset. The final weights are in the range $\left[\frac{1}{1+e}, \frac{1}{1+e^{-1}}\right]$.

$$v_{norm}^{(i)} = 2.v^{(i)} - 1, \; i = 1, 2, \ldots, N \tag{12}$$

$$w^{(i)} = \frac{1}{1 + exp(-v_{norm}^{(i)})}, \; i = 1, 2, \ldots, N \tag{13}$$

During the MAML pretraining phase, the mean weight of each batch is used to weigh the corresponding loss function.

2) Cosine similarity-based measurements

Here we first estimate the cluster center for each target language c_t in the fastText embedding space using the text corpus from AI4Bharat IndicCorp [60].

$$\mathbf{c}_t = mean\left\{\mathbf{c}_t^{(j)}, j = 1, 2, \ldots, N_t\right\}, t = 1, 2, \ldots, 5. \tag{14}$$

where $\mathbf{c}_t^{(j)}$ represents the fastText embedding for the j^{th} sentence from the t^{th} target language and N_t is the total number of sentences in the text corpus from that target language. The number of target languages is 5 in our case. The cosine similarity of the fastText embedding for each source transcription $\mathbf{s}^{(i)}$ with each target language cluster center $\mathbf{c}_t$ is calculated using Equation 6. The overall similarity of the source transcription to the set of target languages $v^{(i)}$ is computed as the mean of the cosine similarities of the source transcription to each of the target language cluster center.

$$v^{(i)} = mean\left\{Cosine\ similarity(\mathbf{s}^{(i)}, \mathbf{c}_t), t = 1, 2, \ldots, 5\right\} \tag{15}$$

The values are normalized to spread over the range [-1, 1] as in Equation 12 and sigmoid of these values are calculated as in Equation 13. The computed values are used to weigh the losses during the MAML pretraining, similar to the case of Mahalanobis distance-based processing.

D. RESULTS AND DISCUSSION

The results of MAML with batch losses being weighted by the similarity of the source to the target language are shown in Table 9. The models are fine-tuned for five epochs in all the cases. Similarity-based weighing provides improvements in all languages except Punjabi. The degradations in Punjabi are minor. With Mahalanobis distance-based similarity weighing, the absolute WER improvements in Marathi, Odia, Sanskrit, and Kannada are 0.9%, 1.9%, 1.4%, and 1.3%, respectively. Cosine similarity-based weights yield absolute WER improvements of 1%, 1.8%, 1.7%, and 0.8% for the above languages. Weighing the losses with text similarity measures yields an average absolute improvement of at least 0.9% in WER. Both similarity measures yield almost the same performance.

Most earlier works on Indian languages report their results on Hindi, Telugu, Gujarati, Bengali, and Tamil, mainly due to the non-availability of off-the-shelf datasets in other languages. In this work, we primarily conduct the experiments in low-resource settings and test our results on recently released datasets in Marathi, Punjabi, Odia, Sanskrit, and Kannada. So, a direct comparison with the earlier works is not possible in most cases. However, Sanskrit Vāksañcayaḥ and Kannada SLR126 datasets have been used in some works previously. [48] uses a multilingual training scheme on transformer architecture and reports a CER/WER of 4.3%/20% in the Sanskrit-OOD subset. Our text-similarity-based weighting scheme has a CER/WER of 5.8%/41.5%. A direct comparison of results is still not possible as they use 56 hours of training data, 50 epochs of fine-tuning and a

monolingual LM during decoding. Our results are with just 10 hours of data, five epochs of fine-tuning, and no explicit language models. Moreover, we test on a 1-hour subset in our experiments. [38] reports a WER of 43.7% in Sanskrit-OOD with monolingual training on time-delay neural networks (TDNNs). [48] reports a CER/WER of 4.1%/22.1% in Kannada with multilingual acoustic and language models trained on 168 hours of annotated speech data from four Dravidian languages. It includes a 49-hour subset of SLR126. We achieve 6.7%/33.6% without language models using just a 10-hour subset of SLR126 and five epochs of fine-tuning. [40] reports a WER of 12.97% on SLR126 using hybrid DNN-HMM architecture and 3-gram language models with subwords as the basic recognition units. However, their models are monolingual and trained on around 275 hours of data.

VII. CONCLUSION

In this work, we established the usefulness of MAML pretraining for building ASR systems quickly for low-resource Indian languages. Since most Indian languages share a common phoneme space, MAML pretraining is much more suited than pretraining with joint multilingual training, as shown by our experiments. MAML exhibits much better adaptation capabilities than joint multilingual training. In the five-epoch fine-tuning setting, MAML beats the joint multilingual pretraining by an absolute 5.4% in CER and 20.3% in WER, on average. MAML also helps in limited data settings where it achieves performances similar to joint multilingual training with just 25% of the data. We showed that language similarities help the target language's ASR performance, and selecting a closely related pretraining language is important in MAML pretraining. We also incorporated a textual similarity-based weighing scheme in the MAML pretraining to achieve average absolute improvements of around 1% in WER.

.

APPENDIX A SLP1 MAPPING SCHEME

SLP1 mapping schemes for the character sets of different Indian languages used in our study are shown in Fig. 11.

REFERENCES

[1] Office of the Registrar General & Census Commissioner India (ORGI), "Census of India 2011 - language atlas - India." [Online]. Available: https://censusindia.gov.in/nada/index.php/catalog/42561/download/46187/Language_Atlas_2011.pdf

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

[3] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2013, pp. 7319–7323.

[4] A. Madhavaraj and A. G. Ramakrishnan, "Data-pooling and multi-task learning for enhanced performance of speech recognition systems in multiple low-resourced languages," in Proc. 25th Nat. Conf. Commun., 2019.

[5] S. Toshniwal et al., "Multilingual speech recognition with a single end-to-end model," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2018, vol. April, pp. 4904–4908.

[6] S. Watanabe, T. Hori, and J. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in Proc. IEEE Workshop Autom. Speech Recognit. Understanding, 2018, vol. January, pp. 265–271.

[7] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," arXiv:1806.05059, Jun. 2018.

[8] S. Kim and M. Seltzer, "Towards language-universal end-to-end speech recognition," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2018, vol. April, pp. 4914–4918.

[9] V. Shetty, M. Sagaya Mary N J, and S. Umesh, "Improving the performance of transformer based low-resource speech recognition for Indian languages," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2020, vol. May, pp. 8279–8283.

[10] D. S. Park et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in Proc. Interspeech, Sep 2019.

[11] C. S. Anoop, A. P. Prathosh, , and A. G. Ramakrishnan, "Unsupervised domain adaptation schemes for building ASR in low-resource languages," in Proc. IEEE Workshop Autom. Speech Recognit. Understanding, 2021, pp. 342–349.

[12] A. Ragni, K. Knill, S. Rath, and M. Gales, "Data augmentation for low resource languages," in Proc. Interspeech, Jan 2014, pp. 810–814.

[13] A. Hannun et al., "Deep speech: Scaling up end-to-end speech recognition," arXiv:1412.5567, 2014.

[14] N. Jaitly and G. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in Proc. Int. Conf. Mach. Learn. - Workshop Deep Learn. for audio, speech, lang. Process., 2013.

[15] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," IEEE Trans. Audio, Speech, Lang. Process., vol. 23, no. 9, pp. 1469–1477, 2015.

[16] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in Proc. Interspeech, 2015.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in Proc. 34th Int. Conf. Mach. Learn., 2017, vol. 3, pp. 1856–1868.

[18] J.-Y. Hsu, Y.-J. Chen, and H.-Y. Lee, "Meta learning for end-to-end low-resource speech recognition," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process, 2020, vol. May, pp. 7844–7848.

[19] S. Singh, R. Wang, and F. Hou, "Improved meta learning for low resource speech recognition," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2022, vol. May, pp. 4798–4802.

[20] A. Naman and K. Deepshikha, "Indic languages automatic speech recognition using meta-learning approach," in Proc. 4th Int. Conf. Natural Lang. Speech Process., 2021, pp. 219–225.

[21] J. Schmidhuber, "Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook," Ph.D. dissertation, Technische Universitat Munchen, Germany, 1987.

[22] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei, "On the optimization of a synaptic learning rule," in Optimality in Art. and Biolog. Neural Net., 1992, pp. 6–8.

[23] M. Andrychowicz et al., "Learning to learn by gradient descent by gradient descent," in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3988–3996.

[24] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in 5th Int. Conf. Learn. Repr., 2017.

[25] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in Proc. 33rd Int. Conf. Mach. Learn., 2016, vol. 4, pp. 2740–2751.

[26] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in Proc. Int. Conf. Mach. Learn., Deep Learn. workshop, 2015.

[27] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. Moura, "Few-shot human motion prediction via meta-learning," Lect. Notes Comput. Sci. (including subseries Lect. Notes Art. Intel. and Lect. Notes Bioinfo.), vol. 11212, pp. 441–459, 2018.

[28] G. Maicas, A. Bradley, J. Nascimento, I. Reid, and G. Carneiro, "Training medical image analysis systems like radiologists," Lect. Notes Comput. Sci. (including subseries Lect. Notes Art. Intel. and Lect. Notes Bioinfo.), vol. 11070, pp. 546–554, 2018.

[29] B. Nguyen, T.-T. Do, B. Nguyen, T. Do, E. Tjiputra, and Q. Tran, "Overcoming data limitation in medical visual question answering," Lect.

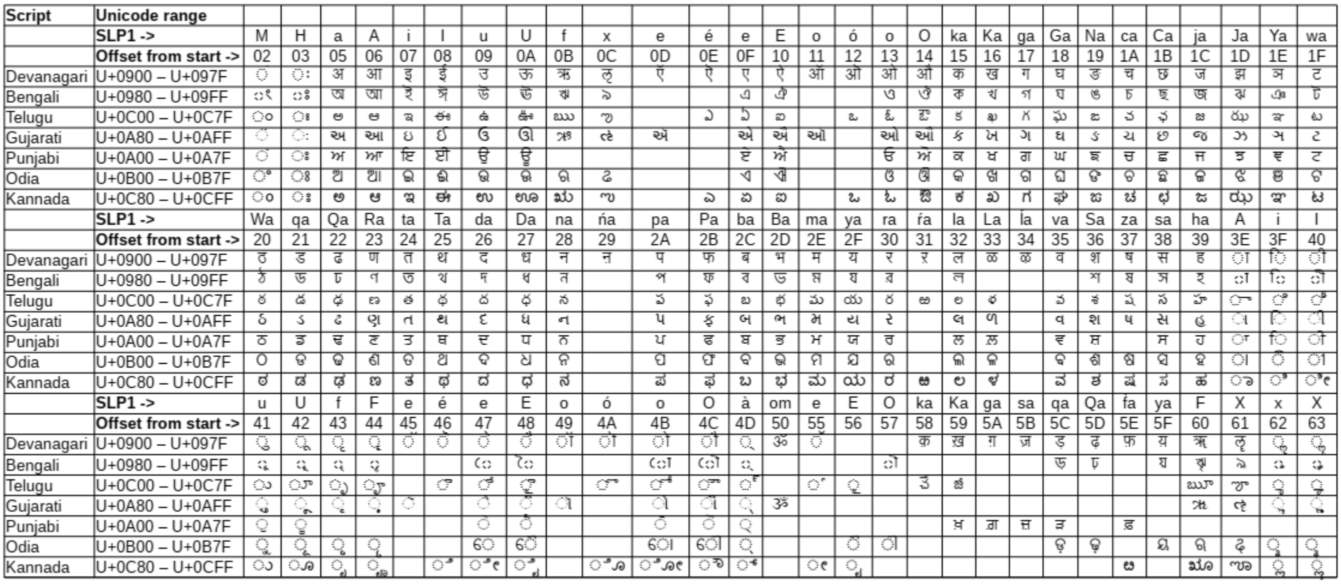


FIGURE 11: SLP1 mapping schemes for the character sets of different Indian languages used in our study. Hindi, Marathi, and Sanskrit use Devanagari script.

Notes Comput. Sci. (including subseries Lect. Notes Art. Intel. and Lect. Notes Bioinfo.), vol. 11767, pp. 522–530, 2019.

[30] G. Winata et al., “Learning fast adaptation on cross-accented speech recognition,” in Proc. Interspeech, 2020, vol. October, pp. 1276–1280.

[31] B. Pulugundla et al., “BUT system for low resource Indian language ASR,” in Proc. Interspeech, Sep 2018, pp. 3182–3186.

[32] S. Khare, A. Mittal, A. Diwan, S. Sarawagi, P. Jyothi, and S. Bharadwaj, “Low resource ASR: The surprising effectiveness of high resource transliteration,” in Proc. Interspeech, 2021, vol. 2, pp. 1051–1055.

[33] A. Diwan et al., “MUCS 2021: Multilingual and code-switching ASR challenges for low resource Indian languages,” in Proc. Interspeech, 2021, vol. 1, pp. 351–355. [Online]. Available: https://www.openslr.org/103/

[34] Gramvaani, “1111 hours Hindi ASR challenge,” Available at https://sites.google.com/view/gramvaaniasrchallenge/home.

[35] Mozilla, “Common voice corpus 10.0.” [Online]. Available: https://commonvoice.mozilla.org/en/datasets

[36] K. S. Bhogale et al., “Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages,” arXiv:2208.12666, 2022. [Online]. Available: https://indic-asr-public.objectstore.e2enetworks.net/shrutilipi/shrutilipi_fairseq.zip

[37] Open-Speech-EkStep, “ULCA ASR dataset corpus.” [Online]. Available: https://storage.googleapis.com/test_public_bucket/external/labelled/Odia_test_set_sme_DD_Odia_01-09-2021_09-43.zip

[38] D. Adiga, R. Kumar, A. Krishna, P. Jyothi, G. Ramakrishnan, and P. Goyal, “Automatic speech recognition in Sanskrit: A new speech corpus and modelling insights,” in Findings Assoc. for Comput. Linguistics, IJCNLP, 2021, pp. 5039–5050. [Online]. Available: https://www.cse.iitb.ac.in/~asr/

[39] A. Madhavaraj, B. Pilar, and A. G. Ramakrishnan, “Subword dictionary learning and segmentation techniques for automatic speech recognition in Tamil and Kannada,” arXiv:2207.13331, 2022. [Online]. Available: https://www.openslr.org/126/

[40] ——, “Knowledge-driven subword grammar modeling for automatic speech recognition in Tamil and Kannada,” 2022.

[41] A. Datta, B. Ramabhadran, J. Emond, A. Kannan, and B. Roark, “Language-agnostic multilingual modeling,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2020, vol. May, pp. 8239–8243.

[42] 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., 2019, vol. May, pp. 5621–5625.

[43] B. Ramani et al., “A common attribute based unified HTS framework for speech synthesis in Indian languages,” in Proc. 8th ISCA Workshop Speech Synth., 2013, pp. 291–296.

[44] P. Scharf, “Linguistic issues and intelligent technological solutions in encoding Sanskrit,” Document Numerique, vol. 16, no. 3, pp. 15–29, 2013.

[45] V. Shetty and S. Umesh, “Exploring the use of common label set to improve speech recognition of low resource Indian languages,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2021, vol. June, pp. 7228–7232.

[46] M. Kumar et al., “Dual script E2E framework for multilingual and code-switching ASR,” in Proc. Interspeech, 2021, vol. 1, pp. 381–385.

[47] C. S. Anoop and A. G. Ramakrishnan, “Investigation of different G2P schemes for speech recognition in Sanskrit,” in Lect. Notes Comput. Sci. (including subseries Lect. Notes Art. Intel. and Lect. Notes Bioinfo.), 2021, vol. 13109 LNCS, p. 536 – 547.

[48] ——, “Exploring a unified ASR for multiple south Indian languages leveraging multilingual acoustic and language models,” in IEEE Spoken Lang. Technol. Workshop, 2022.

[49] ——, “Automatic speech recognition for Sanskrit,” in Proc. 2nd Int. Conf. Intel. Comput., Instrument. Control Technol., 2019, pp. 1146–1151.

[50] M. Choudhury, A. Basu, and S. Sarkar, “A diachronic approach for schwa deletion in Indo Aryan languages,” in Current Themes in Computational Phonology and Morphology: Proc. 7th Meeting ACL Special Interest Group in Comput. Phonology, 2004, pp. 20–26.

[51] X. Chang et al., “An exploration of self-supervised pretrained representations for end-to-end speech recognition,” in Proc. IEEE Workshop Autom. Speech Recognit. Understanding, 2021, p. 228 – 235.

[52] T. Javed et al., “Towards building ASR systems for the next billion users,” in Proc. AAAI Conf. Art. Intel., 2022.

[53] A. Gulati et al., “Conformer: Convolution-augmented transformer for speech recognition,” in Proc. Interspeech, 2020, pp. 5036–5040.

[54] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, vol. 30.

[55] A. Graves, S. Fernández, and F. Gomez, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in Proc. Int. Conf. Mach. Learn., 2006, pp. 369–376.

[56] S. Watanabe, T. Hori, S. Kim, J. Hershey, and T. Hayashi, “Hybrid CTC/attention architecture for end-to-end speech recognition,” IEEE J. Selected Topics Signal Process., vol. 11, no. 8, pp. 1240–1253, 2017.

[57] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Repr., 2015.

[58] S. Watanabe et al., “ESPNet: End-to-end speech processing toolkit,” in Proc. Interspeech, 2018, pp. 2207–2211.

[59] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” arXiv:1607.04606, 2016.

[60] D. Kakwani et al., “IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages,” in Findings Assoc. for Comput. Linguistics, EMNLP, 2020, pp. 4948–4961.

[61] L. McInnes and J. Healy, “UMAP: Uniform manifold approximation and projection for dimension reduction,” arXiv:1802.03426, 2018.

[62] S. Bussey, "Languages in India - how many are there?" Available at https://blog.andovar.com/languages-in-india-how-many-are-there.
[63] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Language id-based training of multilingual stacked bottleneck features," in Proc. Interspeech, 2014, pp. 1–5.
[64] S. Thomas, K. Audhkhasi, J. Cui, B. Kingsbury, and B. Ramabhadran, "Multilingual data selection for low resource speech recognition," in Proc. Interspeech, 2016, vol. September, pp. 3853–3857.
[65] Y. Qian and Z. Zhou, "Optimizing data usage for low-resource speech recognition," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 30, pp. 394–403, 2022.
[66] P. Zhou, Y. Zou, X.-T. Yuan, J. Feng, C. Xiong, and S. Hoi, "Task similarity aware meta learning: Theory-inspired improvement on MAML," in Proc. 37th Conf. Uncert. Art. Intel., 2021, pp. 23–33.
[67] S. T. Jose and O. Simeone, "An information-theoretic analysis of the impact of task similarity on meta-learning," in Proc. IEEE Int. Sympo. Info. Theory, 2021, vol. July, p. 1534 – 1539.

ANOOP C. S. received the Bachelor of Technology from the College of Engineering, Trivandrum, India, and the Master of Engineering from the Indian Institute of Science, Bangalore, India. He is working towards his Ph.D. in Medical Intelligence and Language Engineering Lab, Department of Electrical Engineering, Indian Institute of Science, Bangalore, India. His research interests include signal processing, deep learning, speech recognition in low-resource settings, and natural language processing. He has previously worked in IBM India Pvt. Ltd., Bangalore, India, and Cognizant Technology Solutions, Chennai, India. He has also served in the post of Scientist B for Govt. of India.

A. G. RAMAKRISHNAN is currently a Professor of electrical engineering at the Indian Institute of Science. He has graduated 35 research students and guided over 94 master's theses. He formulated the problem of script recognition at the level of words in printed documents and proposed attention-feedback segmentation of the individual symbols from online handwritten words. He led the National Research Consortium on handwriting recognition in eight languages and was one of the coordinators of the consortium on document image understanding in 12 scripts. Visually impaired students are using over 600 Braille books in Tamil, converted from printed books using his Mozhi Vallaan OCR, which earned him the Manthan Award 2014 in the category e-inclusion and accessibility. He developed unrestricted vocabulary, handwritten word recognition systems for Tamil and Kannada. He proposed a new algorithm for pitch synchronous pitch modification using DCT in the source domain. The Tamil and Kannada TTS developed by him were used by blind students, for which he received the Manthan Award 2015 in the e-education category. His Kannada OCR and TTS were evaluated to be better than Google's Tesseract OCR and Wavenet TTS. He conceived of Linguistic Data Consortium for Indian Languages, currently managed by CIIL, Mysore. He is a member of the FICCI—Indian Language Internet Alliance. He was the President of the Biomedical Engineering Society of India. He is a fellow of the Indian National Academy of Engineering. For his earlier work on nerve conduction in leprosy, he received Sir Andrew Watt Kay Young Researchers Award from the Royal College of Physicians and Surgeons, Glasgow. He is an Associate Editor for Frontiers in Neuroscience—Brain Imaging Methods.