EXPLORING A UNIFIED ASR FOR MULTIPLE SOUTH INDIAN LANGUAGES LEVERAGING MULTILINGUAL ACOUSTIC AND LANGUAGE MODELS

Anoop C S, A G Ramakrishnan

Indian Institute of Science, Bengaluru, India

ABSTRACT

We build a single automatic speech recognition (ASR) model for several south Indian languages using a common set of intermediary labels, which can be easily mapped to the desired native script through simple lookup tables and a few rules. We use Sanskrit Library Phonetic encoding as the labeling scheme, which exploits the similarity in pronunciation across character sets of multiple Indian languages. Unlike the general approaches, which leverage common label sets only for multilingual acoustic modeling, we also explore multilingual language modeling. Our unified model improves the ASR performance in languages with limited amounts of speech data and also in out-of-domain test conditions. Also, the model performs reasonably well in languages with good representation in the training data.

Index Terms— ASR, multilingual acoustic model, multilingual language model, low resourced language, transformer, conformer, Kannada, Telugu, Sanskrit.

1. INTRODUCTION

India has a vast number of living languages, most of which are low-resourced since they do not have enough transcribed speech data to build an automatic speech recognition (ASR) system with good performance. In addition, it is tough to find enough text data to build language models in some of these languages. Some languages are only spoken, with no script. Building acoustic and language models are extremely challenging in such cases due to the scarcity of data. However, there are some opportunities as well. Many of these languages share a common phoneme space and there is one-to-one correspondence between their character sets and pronunciation. Recent work on multilingual speech recognition [1, 2, 3, 4, 5, 6] try to exploit these aspects by combining the small amounts of speech data from multiple languages, thereby minimizing the negative impacts of data scarcity. Another direction of research is to utilize large volumes of unlabelled speech from various languages for unsupervised pretraining followed by the finetuning for the downstream task using labeled data [7, 8, 9]. Generally, these approaches exploit the similarity between languages only in the acoustic models (AM) and use monolingual language models (LM) during decoding. The use of multilingual LM in ASR [10] is not explored much.

We employ multilingual training to build a unified ASR model by pooling speech and text data from several south Indian languages. We convert the text of these languages to a common transcription format using the Sanskrit library phonetic (SLP1) encoding [11] scheme. We show that the unified model with a multilingual LM improves the ASR performance in languages with limited speech or out-of-domain test settings. A language-specific LM can further enhance the results, though.

SLP1 scheme has already been used for speech recognition in Sanskrit [12, 18]. We extend the scheme to multiple south Indian languages utilizing the similarities in pronunciation across their character sets. The idea of using a common set of tokens for multilingual training has already been attempted in [19, 20, 21, 22, 23]. [19] and [20] employ transliteration transducer to transform all languages to Latin script. But training such a transducer requires a lexicon to map the words in the native script to possible Lain script romanizations, necessitating the help of an expert. Some of the languages used in [21] have schwa deletion property, which necessitates language-specific machine translation models for conversion from the common label set to the native script. However, the languages used in our study are free from schwa deletion and allow lookup table-based conversion to the native script. Also, with SLP1, the network's output can be transliterated to any desired script. Such benefits come in handy for spoken languages like Tulu, which uses Kannada or Malayalam script based on the geographical location of the speakers. Audio-to-Byte (A2B) scheme proposed in [22] represents text as a sequence of Unicode bytes. This scheme restricts the output vocabulary size to 256 and shows improvements in multilingual training on languages with single and multi-byte graphemes. The output vocabulary size with SLP1 is 58, much smaller than the A2B scheme. Also, we hope SLP1 can provide better data pooling than the schemes employing a union of graphemes from the constituent languages at the output layer. The use of common to-

978-1-6654-7189-3/22/$31.00 ©2023 IEEE

		Duration (hours)				# Utterances			
Language	**Source**	**train**	**dev**	**test**	**OOD**	**train**	**dev**	**test**	**OOD**
Sanskrit (sa)	Vāksañcayaḥ [12]	56	7	11	5	34309	3337	5529	2778
Telugu (te)	MUCS 2021 [1]	40	5	4		44882	3040	2549	
Kannada (kn)	SLR 126*	49	6	6		14731	1856	1838	
Malayalam (ml)	IITM [13], IIITH [14], SMC [15], SLR 63 [16], CV [17]	23	1	1		15266	1150	1155	
Total		168	19	22	5	109188	9383	11071	2778

* Only a 49-hour subset of SLR 126 is used.

Table 1. Details of the speech corpora used for building and evaluating the multilingual ASR.

Language	# Sentences	# Tokens (native)
Sanskrit (sa)	0.05 M	80
Telugu (te)	0.94 M	82
Kannada (kn)	0.63 M	78
Malayalam (ml)	0.86 M	86

Table 2. Details of the text corpora taken from wikipedia for training the monolingual language models.

kens across languages for acoustic modeling in our work is closer to the scheme in [23]. But we extend the use of tokens to multilingual LM also.

A limitation of the proposed approach is that it cannot cope with code-switching between the languages under study, as there is no way to detect the switching and hence no way to map back to the correct script.

2. DATASETS USED FOR THE STUDY

We use publicly available speech data from four south Indian languages - Telugu, Kannada, Malayalam, and Sanskrit. Each dataset has three subsets generally - train, dev, and test. Sanskrit has an additional split called *out of domain* (OOD), which contains speech data from speakers with distinguishable influence of their native language and from domains that are not part of the training set [12]. For Malayalam, we combine speech data from four publicly available datasets, namely IITM [13], IIITH [14], SMC [15] and SLR63 [16] to form the training set. Common voice (CV) [17] corpus is split to form dev and test subsets. The details of the datasets are shown in Table 1. Note that Malayalam has only around 23 hours of speech data for training. The unified multilingual acoustic model employs around 168 hours of annotated speech data for training.

We use wiki data dumps in each language as the text corpus for building LMs. The details of the text corpora are shown in Table 2. The number of native script tokens used to train monolingual LMs in each language are shown in the last column. These include the additional tokens like *blank*, *unk* and *sos/eos*. The Sanskrit text corpus is quite small in size.

3. EXPERIMENTAL SETUP

3.1. Conversion to SLP1 labels

In contrast with English, most Indian languages possess a one-to-one correspondence between the literals in the native script and their pronunciation, with a few exceptions like the anusvara and visarga characters, which can be handled easily with simple rules. The Unicode tables for Indian languages are organized such that the letters with similar pronunciation occur at the same offset from the beginning of the assigned range for every language. Hence, mapping the characters from different scripts to a common label set is simple for Indian languages. We use the SLP1 scheme [11] for this. It maps both a vowel and its modifier to the same ASCII character. We modify these mappings slightly to incorporate the general rules of pronunciation for anusvara and visarga as described in [24]. These modifications have been shown to be useful in improving the speech recognition performance in [18]. Figure 1 shows the SLP1 mappings for the languages under our study, which are all free from schwa deletion. Hence, the mapping of SLP1 literals back to the native script is easily achieved using lookup tables. There are a few exceptions for characters like vowel modifiers, anusvara, and visarga. However, we handle them easily with simple rules.

3.2. Model architecture of the unified ASR

The architecture of the unified ASR model employed in this work is shown in Fig. 2. We use the upstream-downstream setup as in [25]. Wav2vec 2.0 [7] based pre-trained model is used as the front-end feature extractor in the upstream. Authors of [26] show that using target domain data during pretraining leads to large performance improvements in ASR. So we use models pretrained on Indian languages to provide crosslingual speech representations from raw audio. Specifically, we use Indic wav2vec [9] large model pre-trained on 17,000 hours of raw speech data from 40 Indian languages. Also, the pretraining includes data from diverse domains such as education, news, technology, and finance. Parameters

SLP1	M	H	a	A	i	I	u	U	f	x	é	e	E	ó	o	O	A	i	I	u	U	f	F	é	e	E	ó	o	O	à
Offset from start	02	03	05	06	07	08	09	0a	0b	0c	0e	0f	10	12	13	14	3e	3f	40	41	42	43	44	46	47	48	4a	4b	4c	4d
Sanskrit (U+0900)	◌ं	◌ः	अ	आ	इ	ई	उ	ऊ	ऋ	ऌ	ऎ	ए	ऐ	ऒ	ओ	औ	◌ा	◌ि	◌ी	◌ु	◌ू	◌ृ	◌ॄ	◌ॆ	◌े	◌ै	◌ॊ	◌ो	◌ौ	◌्
Telugu (U+0c00)	◌ం	◌ః	అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ఌ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	◌ా	◌ి	◌ీ	◌ు	◌ూ	◌ృ	◌ౄ	◌ె	◌ే	◌ై	◌ొ	◌ో	◌ౌ	◌్
Kannada (U+0c80)	◌ಂ	◌ಃ	ಅ	ಆ	ಇ	ಈ	ಉ	ಊ	ಋ	ಌ	ಎ	ಏ	ಐ	ಒ	ಓ	ಔ	◌ಾ	◌ಿ	◌ೀ	◌ು	◌ೂ	◌ೃ	◌ೄ	◌ೆ	◌ೇ	◌ೈ	◌ೊ	◌ೋ	◌ೌ	◌್
Malayalam (U+0d00)	◌ം	◌ഃ	അ	ആ	ഇ	ഈ	ഉ	ഊ	ഋ	ഌ	എ	ഏ	ഐ	ഒ	ഓ	ഔ	◌ാ	◌ി	◌ീ	◌ു	◌ൂ	◌ൃ	◌ൄ	◌െ	◌േ	◌ൈ	◌ൊ	◌ോ	◌ൌ	◌്

SLP1	ka	Ka	ga	Ga	Na	ca	Ca	ja	Ja	Ya	wa	Wa	qa	Qa	Ra	ta	Ta	da	Da	na	pa	Pa	ba	Ba	ma	ya	ra	ŕa	la	La	ĺa	va	Sa	za	sa	ha
Offset from start	15	16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24	25	26	27	28	2a	2b	2c	2d	2e	2f	30	31	32	33	34	35	36	37	38	39
Sanskrit (U+0900)	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ	ड	ढ	ण	त	थ	द	ध	न	प	फ	ब	भ	म	य	र	ऱ	ल	ळ	ऴ	व	श	ष	स	ह
Telugu (U+0c00)	క	ఖ	గ	ఘ	ఙ	చ	ఛ	జ	ఝ	ఞ	ట	ఠ	డ	ఢ	ణ	త	థ	ద	ధ	న	ప	ఫ	బ	భ	మ	య	ర	ఱ	ల	ళ		వ	శ	ష	స	హ
Kannada (U+0c80)	ಕ	ಖ	ಗ	ಘ	ಙ	ಚ	ಛ	ಜ	ಝ	ಞ	ಟ	ಠ	ಡ	ಢ	ಣ	ತ	ಥ	ದ	ಧ	ನ	ಪ	ಫ	ಬ	ಭ	ಮ	ಯ	ರ	ಱ	ಲ	ಳ		ವ	ಶ	ಷ	ಸ	ಹ
Malayalam (U+0d00)	ക	ഖ	ഗ	ഘ	ങ	ച	ഛ	ജ	ഝ	ഞ	ട	ഠ	ഡ	ഢ	ണ	ത	ഥ	ദ	ധ	ന	പ	ഫ	ബ	ഭ	മ	യ	ര	റ	ല	ള	ഴ	വ	ശ	ഷ	സ	ഹ

Fig. 1. SLP1 mapping scheme for the languages used in this work.

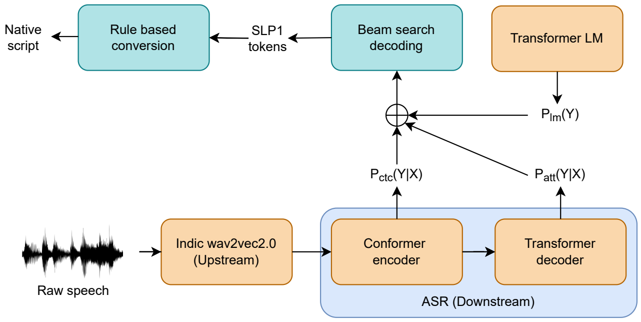

Fig. 2. The architecture of the proposed end-to-end ASR system, with multilingual acoustic and language models.

Parameters	Acoustic model (Conformer)	Language Model (Transformer)
Embedding dim.	-	128
Kernel size	15	-
Encoder layers	12	16
Decoder layers	6	-
Attention heads	4	8
Attention dim.	256	512
Feed-forward dim.	2048	2048

Table 3. Hyperparameter values used for acoustic and language models.

of the upstream model are frozen during the training of ASR. The Indic wav2vec-large model outputs 1024-dimensional vectors at the output, which are converted into 80-dimensional features by a linear layer. These features are fed to the downstream ASR model.

We use conformers [27] for acoustic modeling in the downstream ASR. The hyperparameters for the conformer model are listed in Table 3. We use the joint CTC-attention scheme [28] for training the model with a CTC weight of 0.3 and an attention weight of 0.7. We use Adam optimizer [29] with a peak learning rate of 0.005 and warmup steps of 30000. We train the model for 50 epochs. Ten best models based on validation accuracy are averaged and used for decoding.

We use transformer [30] encoder structure for building LMs. The hyperparameters for the LMs are listed in Table 3. We train the model for a maximum of 50 epochs with a patience value of 10. The best six models based on validation loss are averaged to form the final model. The optimizer is Adam, with a peak learning rate of 0.001 and warmup-steps of 25000. During decoding, the most probable token sequence $\hat{Y}$ is computed as:

$$\hat{Y} = \underset{Y \in V^*}{\arg\max} \Big[\lambda \, log \; p_{ctc}(Y|X) + (1-\lambda) log \; p_{att}(Y|X) \\ + \gamma \, log \; p_{lm}(Y)\Big] \quad (1)$$

where p_{ctc}, p_{att}, and p_{lm} are the CTC, attention, and language model scores, respectively. λ and γ are the CTC and language model weights and V^* is the set of all target hypotheses. We use beam size of 20, $\lambda = 0.3$, and $\gamma = 1.2$. All the experiments are conducted using ESPNet toolkit [31].

3.3. Details of baseline and multilingual models

In the baseline setup, we create acoustic and language models using only the data from the corresponding language. We refer to these models as *mono*. Here we use the text in the native script. The number of tokens in the native script for each language is shown in Table 2. The baselines use different acoustic and language models for each language.

For building the multilingual AM, we combine the speech data from the train sets of all four languages. The corresponding text transcriptions in the native script are converted to SLP1. SLP1 uses 58 tokens for the multilingual text representation. Similarly, the text corpora from multiple languages are converted to SLP1 format to build the multilingual LM. The SLP1 outputs of these models are converted to the native script using lookup tables and some rule-based modifications. We evaluate the performance of these multilingual AM and LM on test/OOD sets for all the languages used in training. We compare these results to the baseline results with monolingual models.

4. RESULTS AND DISCUSSION

The performances of both mono and multilingual models are evaluated based on character error rate (CER) and word error rate (WER).

4.1. Performance of baseline monolingual models

The baseline results obtained with monolingual models trained on the native script are shown in Table 5. We delineate the results on Malayalam and Sanskrit with a line since limited audio resources (≈ 23 hours) were used in training the Malayalam AM, and limited text resources (≈ 0.05 M sentences) were used in building the Sanskrit LM. The Sanskrit *OOD* set has data from domains completely different from those used in training. Speakers forming the *OOD* set have a noticeable influence of their native language on their speech.

The agglutinative nature of the languages under study often results in long compound words. However, while uttering these words, the speakers usually pause at arbitrary points. The splitting of these words may not be driven by *sandhi* rules depending on the speaker's expertise in the language. So the short pauses in utterances may not indicate word endings. This fact exaggerates the CER and WER while estimating the performance of ASR on these languages. Table 4 shows an example. Due to the arbitrary pause by the speaker, the long word सस्यश्यामलाम् is decoded as two words. Though the ASR output is perfect in this case, the WER measure penalizes it with two errors. CER is also penalized by an insertion error due to the extra space. We ignore such errors and recompute the CER and WER in the column *Ignore space errors*. However, we do not handle merges and splits according to the *sandhi* rules, where the graphemes at the boundary morph into different ones.

Type	Sentence
Reference :	सस्यश्यामलाम्
Decoded by ASR :	सस्य श्यामलाम्
WER Evaluation :	1 substitution + 1 insertion

Table 4. An example showing the degradation in WER due to the arbitrary pauses introduced by the speaker.

Setup S1 of Table 5 reports the results of monolingual acoustic models without any language models. Setup S2 uses monolingual LMs trained on the resepective native scripts for decoding. The best result among the setups for each dataset is highlighted in bold. We can see that incorporating monolingual LM trained on the native script improves the performance of ASR across all the test sets except Sanskrit-test. The Sanskrit-test set contains data from domains similar to the ones used to train the AMs. However, the text corpus for building the Sanskrit LM includes data from more generic domains than the training data and has a limited size. This could be the reason for the performance dip on Sanskrit. However, the OOD set in Sanskrit benefits from the incorporation of LM. After ignoring the space errors explained in Table 4, CER and WER improvements in Sanskrit-OOD set are 0.4% and 3.9%, respectively. Malayalam-test set shows improvements of 0.5% and 2.4% in CER and WER, respectively, in setup S2.

Setup	Test set	LM	Native script		Ignore space errors	
			CER	WER	CER	WER
S1	te - test	-	9.4	36.8	9.1	33.4
	kn - test	-	4.7	25.8	4.3	21.0
	sa - test	-	**2.4**	**16.3**	**2.0**	**11.7**
	sa - OOD	-	5.6	36.0	4.9	26.4
	ml - test	-	10.3	49.1	9.8	44.7
S2	te - test	mono	**6.5**	**23.6**	**6.2**	**19.7**
	kn - test	mono	**3.9**	**20.2**	**3.5**	**15.8**
	sa - test	mono	2.6	**16.3**	2.3	12.8
	sa - OOD	mono	**5.2**	**31.1**	**4.5**	**22.5**
	ml - test	mono	**9.9**	**47.2**	**9.3**	**42.3**

Table 5. Results of the baseline monolingual acoustic models trained using native script on the test/OOD datasets of languages used in training. The best results in each evaluation set are shown in bold.

In Indian languages, the word boundaries can vary, and speakers can split compound words at arbitrary positions and merge simple words to form compound words. By ignoring these space errors, the WER improves by an average of 5.4% in the S1 setup and 5.1% in the S2 setup. CER also improves by an average of 0.5% in both setups. These improvements indicate the extent of overestimation of WER and CER in Indian languages. The speakers may also split/merge words according to the *sandhi* rules causing sound changes at the morpheme or word boundaries, which is not captured in the reference transcription. We can further reduce the exaggeration in CER/WER by treating the sandhi-based merges and splits as valid. However, we do not handle it here since it requires inputs from expert linguists.

4.2. Results of unified multilingual model

Table 6 lists the results with the unified multilingual model trained on SLP1 text. There are 3 setups: i) S3: no LM, ii) S4: monolingual LM trained on SLP1 text data of respective languages, and iii) S5: multilingual LM trained on SLP1 text data from all the languages. The acoustic model is the same for all languages in all the setups. Comparing setup S3 with S1, the equivalent setup in the baseline, we see that the performance of the common multilingual AM matches reasonably well with that of language-specific AMs. On Malayalam-test and Sanskrit-OOD sets, CER matches with the S1 results. CER (WER) decreases by 0.5% (2.7%) on Telugu. CER increases by 0.5% on Sanskrit, and 0.3% on Kannada test sets. On Sanskrit OOD set, WER reduces by 1.4%. There is a slight increase of 0.7% in WER on Malayalam-test set, where only limited speech data is available for AM training. On Sanskrit and Kannada test sets, WER increases by 1.3% and 4.8%, respectively.

Joint decoding with the unified multilingual AM and monolingual LM trained from SLP1 transcriptions (setup S4) decisively improves the performance over the S3 setup on all sets except Sanskrit-test. This trend is similar to the performance improvements in the setup S2 over S1. Specifically, CER (WER) improves by 2.6% (10.6%) for Telugu, 0.4% (3.2%) for Kannada, 0.5% (4.6%) for Sanskrit-OOD and by 6.4% (28.2%) for Malayalam. The improvements are huge for Malayalam, which has limited representation in the multilingual AM training but reasonably good amount of text data for LM training. However, the results degrade marginally for Sanskrit-test. Decoding with an LM trained on a text corpus from rather generic domains seems to be the cause of this degradation, since Sanskrit-test contains data from domains similar to the oness in the train set.

The S5 setting uses joint decoding with the unified multilingual AM and the multilingual LM trained on SLP1 text from all the languages. The benefit of this scheme is that there are only single AM and LM models for all the languages in the study. The ASR performance improves over S3 on all sets except Sanskrit-test. CER (WER) improves by 2.2% (9.4%), 0.5% (3.7%), 2.1% (12.1%) and 0.4% (3.8%), respectively, for Telugu, Kannada, Malayalam test and Sanskrit-OOD sets. CER (WER) degrades by 0.4% (1.4%) for Sanskrit-test. WER improves over the baseline setup S2 for Malayalam-test (9%) and Sanskrit-OOD (1.3%). However, WER degrades over S2 for reasonably rich resourced languages (1.6% in Telugu and Sanskrit, and 6.3% in Kannada test sets) as the AM and LM are made more generic through multilingual training. The results are more consistent across languages in S5 than in S2. Joint decoding with multilingual AM and LM seems to help in cases where:

(i) the amount of transcribed speech data available for multilingual training is lower than that from other languages (as seen on the Malayalam-test set).

(ii) the text data for building language models are limited, and the test set is from unseen domains (as seen on the Sanskrit-OOD set).

WER performance of S5 degrades slightly in most languages (1.2% in Telugu-test, 0.5% in Sanskrit-test, and 0.8% in Sanskrit-OOD) compared to the language-specific LMs in S4. In Malayalam, degradations are much higher (16.1%). Recognition is largely dependent on the performance of LM in Malayalam since its data representation in AM training is much less than the other languages. As the language models become more generic in S5, performance degrades heavily in Malayalam. However, S5 improves over S4 by 0.5% on Kannada-test.

Setup	Test set	LM	SLP1 script		Native script		Ignore space errors	
			CER	WER	CER	WER	CER	WER
S3	te - test	-	8.0	32.3	8.9	34.3	8.6	30.7
	kn - test	-	3.2	24.5	5.0	30.7	4.6	25.8
	sa - test	-	**2.0**	15.3	**2.8**	17.2	**2.5**	**13.0**
	sa - OOD	-	4.5	32.8	5.5	34.6	4.9	25.0
	ml - test	-	8.8	48.6	10.1	49.8	9.8	45.4
S4	te - test	mono	**5.5**	**21.7**	**6.3**	**23.9**	**6.0**	**20.1**
	kn - test	mono	**2.8**	**20.9**	4.5	26.5	4.2	22.6
	sa - test	mono	2.1	**14.4**	3.0	**16.4**	2.9	13.9
	sa - OOD	mono	**4.0**	**27.2**	**5.0**	**29.1**	**4.4**	**20.4**
	ml - test	mono	**2.6**	**18.1**	**3.6**	**19.6**	**3.4**	**17.2**
S5	te - test	multi	5.8	22.9	6.6	25.0	6.4	21.3
	kn - test	multi	**2.8**	**20.9**	**4.4**	**25.8**	**4.1**	**22.1**
	sa - test	multi	2.1	15.2	3.1	17.2	2.9	14.4
	sa - OOD	multi	**4.0**	28.1	5.1	30.1	4.5	21.2
	ml - test	multi	6.8	36.4	8.0	37.9	7.7	33.3

Table 6. Results of multilingual models trained using SLP1 tokens on test/OOD datasets of languages used in training. The best results for each dataset are highlighted in bold.

Among the unified multilingual AMs, the setup S4 seems to be the best. However, it requires a reasonable amount of text data to build language-specific LMs, which is not readily available in many spoken languages.

4.3. Results of retraining the models

We further retrain the unified multilingual model with monolingual speech to create language-specific AMs. We expect the pretraining to provide better initialization for the language-specific models. The training is performed using the SLP1 script. We evaluate the performance of all the models in 3 settings: i) S6: no LM, ii) S7: with a monolingual LM trained on SLP1 text data from the individual languages, and iii) S8: with a monolingual LM obtained by retraining the multilingual LM of S5 with the SLP1 text data from individual languages (referred to as *retrain*). Table 7 lists the results on the evaluation sets using retrained AMs. Comparing the setup S6 with its counterpart S3 in the unified multilingual model, we see that retraining does not change the ASR performance much. The improvements/degradations are limited to 0.2% in CER and 1.1% in WER.

Joint decoding with retrained AM and SLP1-trained monolingual LM (S7) greatly improves over S6 on all the datasets. WER improvements are 10.1%, 3.1%, 0.5%, 5.7%, and 12%, respectively, for Telugu-test, Kannada-

Setup	Test set	LM	SLP1 script		Native script		Ignore space errors	
			CER	WER	CER	WER	CER	WER
S6	te - test	-	7.8	31.9	8.6	33.8	8.4	30.1
	kn - test	-	3.1	24.1	4.9	29.8	4.6	25.3
	sa - test	-	2.1	16.9	3.0	18.7	2.7	14.1
	sa - OOD	-	4.6	33.5	5.7	35.2	5.0	25.7
	ml - test	-	8.7	48.6	10.1	49.8	9.8	45.3
S7	te - test	mono	5.6	**21.6**	6.4	**23.8**	6.1	**20.0**
	kn - test	mono	**2.7**	**20.7**	4.4	26.2	4.1	22.2
	sa - test	mono	2.0	14.5	3.0	16.4	2.8	13.6
	sa - OOD	mono	**3.9**	**27.1**	**4.9**	**29.0**	**4.3**	**20.0**
	ml - test	mono	6.8	37.0	8.0	38.3	7.7	33.3
S8	te - test	retrain	**5.4**	**21.1**	**6.2**	**23.3**	6.0	**19.7**
	kn - test	retrain	**2.7**	**20.5**	4.4	26.0	4.2	22.2
	sa - test	retrain	2.0	14.5	3.0	16.4	2.8	13.6
	sa - OOD	retrain	**3.9**	**27.1**	**4.9**	**29.0**	**4.3**	**20.0**
	ml - test	retrain	6.3	34.3	7.3	35.5	7.1	30.8

Table 7. Results of retraining the multilingual acoustic model with SLP1 transcriptions from each language. We have a separate AM for each language here. The numbers in bold indicate the improvements over the best results with the unified multilingual model.

test, Sanskrit-test, Sanskrit-OOD and Malayalam-test sets. However, the improvements over S4 are meager, if any. Retraining the LMs helps only Malayalam, where the CER (WER) improves by 0.6% (2.5%). The results of retrained LMs (S8) on other languages are almost identical to those of monolingual LMs (S7). The results of all the setups are summarized in Figs. 3 and 4.

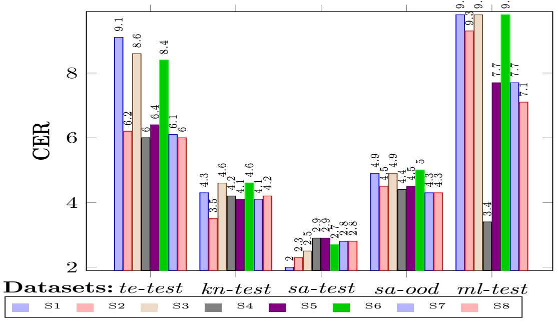

Fig. 3. Comparison of CER in different setups.

4.4. Performance on unseen languages

We test our unified multilingual model on two languages unseen during training, namely Tulu and Tamil. We use 826 Tulu utterances (≈ 35 min.) from youtube and 2609 Tamil utterances (≈ 4 hrs) from the MUCS 2021 [1] test set. The results are listed in Table 8. Though Tulu is an independent language, it has a few similarities with Kannada, which is used in training acoustic models. Tamil is entirely different from the languages used in training. It does not have separate letters for aspirated and voiced stops. For eg. /*ka*/, /*kha*/, /*ga*/, and /*gha*/ are all mapped to /*ka*/ (க) and context-based phonetic rules decide the pronunciation. The number of valid SLP1 tokens are just 40 in Tamil. Hence the unified multilingual model has better results on Tulu than on Tamil. The multilingual language model, which does not have any representation from these languages, worsens the results further.

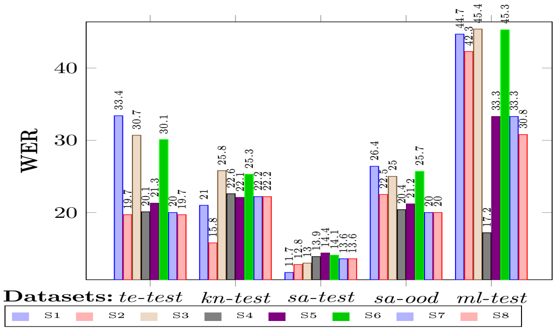

Fig. 4. Comparison of WER in different setups.

Test set	LM	SLP1 script		Native script		Ignore space errors	
		CER	WER	CER	WER	CER	WER
Tulu	-	15.9	51.2	16.1	51.5	16.0	50.2
Tamil	-	30.7	85.0	30.7	85.2	30.6	84.1
Tulu	multi	38.8	105.1	36.8	101.4	36.7	100.8
Tamil	multi	44.9	109.5	43.0	106.8	42.9	106.3

Table 8. Results of multilingual AM on languages unseen during training.

5. CONCLUSIONS

We explore the use of unified acoustic and language models for ASR in several south Indian languages with the help of a common intermediary labeling scheme called SLP1. The advantage of such a system is its simplicity, since no language ID component is needed during training or inference. With the help of a lookup table, we can easily convert the SLP1 output of the model to any desired script. Languages like Tulu without a script of their own (and use the scripts of one or more other languages), can benefit from this scheme. The system betters the baseline performance on Malayalam-test, where only limited speech data is available for training the acoustic models, and on Sanskrit-OOD, where the speech data is from domains different from that used in training. However, multilingual AM + LM does not outperform monolingual AM + LM for rich resource languages. Multilingual AM gives similar results with mono and multilingual LMs in all languages except Malayalam.

6. ACKNOWLEDGMENT

We thank SERB, Govt. of India for partially funding this research through IMPRINT2 project, IMP/2018/000504.

7. REFERENCES

[1] A. Diwan, R. Vaideeswaran, S. Shah, A. Singh, S. Raghavan, S. Khare, V. Unni, S. Vyas, A. Rajpuria, C. Yarra, A. Mittal, P. Ghosh, P. Jyothi, K. Bali, V. Seshadri, S. Sitaram, S. Bharadwaj, J. Nanavati, R. Nanavati, and K. Sankaranarayanan, "MUCS 2021: Multilingual and code-switching ASR challenges for low resource Indian languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, vol. 1, 2021, pp. 351–355.

[2] O. Klejch, E. Wallington, and P. Bell, "The CSTR system for multilingual and code-switching ASR challenges for low resource Indian languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, vol. 2, 2021, pp. 1001–1005.

[3] A. Madhavaraj and A. G. Ramakrishnan, "Data and knowledge-driven approaches for multilingual training to improve the performance of speech recognition systems of Indian languages," *arXiv preprint arXiv:2201.09494*, 2022.

[4] S. Abate, M. Tachbelie, and T. Schultz, "End-to-end multilingual automatic speech recognition for less-resourced languages: The case of four Ethiopian languages," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process., ICASSP*, vol. June, 2021, pp. 7013–7017.

[5] 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 *2019 National Conf. Commun. (NCC)*. IEEE, 2019, pp. 1–5.

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

[7] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Adv. Neural Inf. Proc. Sys.*, 2020.

[8] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, vol. 1, 2021, pp. 346–350.

[9] T. Javed, S. Doddapaneni, A. Raman, K. S. Bhogale, G. Ramesh, A. Kunchukuttan, P. Kumar, and M. M. Khapra, "Towards building ASR systems for the next billion users," in *Proc. of the AAAI Conf. on Artificial Intelligence*, 2022.

[10] C. Fügen, S. Stüker, H. Soltau, F. Metze, and T. Schultz, "Efficient handling of multilingual language models," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*, 2003, pp. 441–446.

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

[12] 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," *Findings of the Assoc. for Comp. Linguistics: ACL-IJCNLP*, pp. 5039–5050, Aug 2021.

[13] TTS Consortium, TDIL, Meity, "Indic TTS Malayalam speech corpus," 2016. [Online]. Available: https://www.kaggle.com/code/mpwolke/indic-tts-malayalam-speech-corpus/data

[14] K. Prahallad, E. Kumar, V. Keri, S. Rajendran, and A. Black, "The IIIT-H Indic speech databases," in *Proc. of 13th Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, vol. 3, 2012, pp. 2545–2548. [Online]. Available: http://www.festvox.org/databases/iiit_voices/iiit_mal_abi.tar.gz

[15] Swathathra Malayalam Computing, "Malayalam speech corpus," 2020. [Online]. Available: https://releases.smc.org.in/msc-reviewed-speech/LATEST/msc-reviewed-speech-v1.0+20200825.zip

[16] F. He, S.-H. C. Chu, O. Kjartansson, C. Rivera, A. Katanova, A. Gutkin, I. Demirsahin, C. Johny, M. Jansche, S. Sarin, and K. Pipatsrisawat, "Open-source multi-speaker speech corpora for building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu speech synthesis systems," in *Proc. of The 12th Lang. Resources and Eval. Conf. (LREC)*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 6494–6503.

[17] Common Voice Corpus 9.0, "Malayalam speech corpus," 2022. [Online]. Available: https://commonvoice.mozilla.org/en/datasets

[18] C. S. Anoop and A. G. Ramakrishnan, "Investigation of different G2P schemes for speech recognition in Sanskrit," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13109 LNCS, 2021, p. 536 – 547.

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

[20] J. Emond, B. Ramabhadran, B. Roark, P. Moreno, and M. Ma, “Transliteration based approaches to improve code-switched speech recognition performance,” in *Proc. IEEE Spoken Lang. Tech. Workshop (SLT)*, 2019, pp. 448–455.

[21] M. Kumar, J. Kuriakose, A. Thyagachandran, A. Kumar, A. Seth, L. Prasad, S. Jaiswal, A. Prakash, and H. Murthy, “Dual script E2E framework for multilingual and code-switching ASR,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, vol. 1, 2021, pp. 381–385.

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

[23] 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., ICASSP*, vol. June, 2021, pp. 7228–7232.

[24] C. S. Anoop and A. G. Ramakrishnan, “Automatic speech recognition for Sanskrit,” in *2nd Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies, ICICICT*, 2019, pp. 1146–1151.

[25] X. Chang, T. Maekaku, P. Guo, J. Shi, Y.-J. Lu, A. S. Subramanian, T. Wang, S.-W. Yang, Y. Tsao, H.-Y. Lee, and S. Watanabe, “An exploration of self-supervised pretrained representations for end-to-end speech recognition,” in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU - Proceedings*, 2021, p. 228 – 235.

[26] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, and M. Auli, “Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, vol. 3, 2021, pp. 2123–2127.

[27] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, 2020, pp. 5036–5040.

[28] 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.

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

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Adv. Neural Inf. Proc. Sys.*, vol. 30, 2017.

[31] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “ESPNet: End-to-end speech processing toolkit,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, 2018, pp. 2207–2211.