Suitability of syllable-based modeling units for end-to-end speech recognition in Sanskrit and other Indian languages

Anoop Chandran Savithri[a] (anoopcs@iisc.ac.in), Ramakrishnan Angarai Ganesan[b] (agr@iisc.ac.in)

[a] Medical Intelligence and Language Engineering Lab, Department of Electrical Engineering, Indian Institute of Science, Bengaluru, Karnataka, 560012, India

[b] Medical Intelligence and Language Engineering Lab, Department of Electrical Engineering, Indian Institute of Science, Bengaluru, Karnataka, 560012, India

Corresponding Author:

Anoop Chandran Savithri

Medical Intelligence and Language Engineering Lab, Department of Electrical Engineering, Indian Institute of Science, Bengaluru, Karnataka, 560012, India

Tel: +919611127851

Email: anoopcs@iisc.ac.in

Suitability of syllable-based modeling units for end-to-end speech recognition in Sanskrit and other Indian languages

Anoop Chandran Savithri[a,*], Ramakrishnan Angarai Ganesan[a]

[a]*Medical Intelligence and Language Engineering Lab, Department of Electrical Engineering, Indian Institute of Science, Bengaluru, Karnataka, 560012, India*

Abstract

Most Indian languages are spoken in units of syllables. However, speech recognition systems developed so far for Indian languages have generally used characters or phonemes as modeling units. This work evaluates the performance of syllable-based modeling units in end-to-end speech recognition for several Indian languages. The text is represented in 3 different forms: native script, Sanskrit library phonetics (SLP1) encoding, and syllables, and tokenized with sub-word units like character, byte-pair encoding (BPE), and unigram language modeling (ULM). The performances of these tokens in monolingual training and cross-lingual transfer learning are compared. Syllable-based BPE/ULM subword units give promising results in the monolingual setup if the dataset is sufficiently diverse to represent the syllable distribution in the language. For the Vāksañcayaḥ dataset in Sanskrit, syllable-BPE tokens achieve state-of-the-art results. The capability of syllable-BPE units to complement SLP1-character models through a pretraining-finetuning setup is also evaluated. For Sanskrit, syllable-BPE achieves better word error rates (WER) than the pretraining-finetuning approaches. For Tamil and Telugu, both result in comparable WERs. SLP1-character units are largely better than syllable-based units for cross-lingual transfer learning.

*Corresponding author.

Email addresses: anoopcs@iisc.ac.in (Anoop Chandran Savithri), agr@iisc.ac.in (Ramakrishnan Angarai Ganesan)

Keywords: automatic speech recognition, end-to-end, Sanskrit, Indian languages, low-resource, syllable-based ASR

1. Introduction

The writing systems in the world can be broadly classified into 1) Alphabets, 2) Syllabaries, and 3) Semanto-phonetic writing systems. Graphemes represent consonants and vowels in alphabets. Most European, Indo-Aryan, and Afro-Asiatic languages fall in this category. Syllabaries use syllables as graphemes as in Japanese Hiragana/Katakana. They have a large number of characters. Symbols represent both sound and meaning in semanto-phonetic writing systems like Chinese. They also have a large number of symbols. There is no theoretical upper limit to the number of symbols in such scripts.

Most of the works on end-to-end (E2E) speech recognition in languages with alphabet-based writing systems use graphemes as the basic units for training (Watanabe et al., 2017a; Gulati et al., 2020; Toshniwal et al., 2018; Anoop & Ramakrishnan, 2021a). There were a few attempts to use phonemes (Zeyer et al., 2020) and syllables (Zhao et al., 2019). E2E automatic speech recognition (ASR) on syllabaries like Japanese also employ graphemes (Karita et al., 2021). However, graphemes themselves are manifestations of syllables in Japanese Hiragana/Katakana. In semanto-phonetic writing systems like Mandarin Chinese, different token units like characters, context-independent phonemes, pinyins (a scheme to transcribe the Mandarin Chinese pronunciations using Roman alphabets), syllables, words, and subwords have been explored (Watanabe et al., 2017a; Zou et al., 2018; Chan & Lane, 2016; Zhang et al., 2019; Yuan et al., 2021; Zhou et al., 2018). Many of these studies show the usefulness of pronunciation-based pinyins and syllables in E2E ASR (Chan & Lane, 2016; Zhang et al., 2019; Yuan et al., 2021). Bytes have also been used as tokens to perform language-independent speech recognition (Li et al., 2019).

Syllables might be a better unit than graphemes for text representation in ASR for Indian languages due to the following reasons:

1. Vowels are represented by original graphemes when they appear at the beginning of a word and with modifiers otherwise. Eg. आत्मा (/a:tma:/ meaning soul) has आ at the beginning and ◌ा in the end, both denoting the stressed (long) vowel "a:". This duplication in graphemes may create some rendering errors during decoding and may exaggerate the actual word error rate (WER).
2. Most of the writing systems of the Indian subcontinent use specific symbols to signify the lack of the inherent vowel /ə/ (schwa). In the earlier example of त्म = त + ◌् + म , ◌् indicates the lack of the inherent vowel. Models need to explicitly learn when to output ◌् from the context.
3. The speakers may split/combine the words at the boundaries according to the *sandhi* rules, and transcriptions may not reflect that. This causes rendering errors in the form of substitution, insertion, and/or deletion.
4. Indian languages are spoken in units of syllables, which are bigger modeling units and may capture the long contexts better. They can also reduce the length of the text representation, which may help in decoding long sequences. However, syllable units increase the vocabulary size resulting in bigger linear and softmax layers at the output. So, it is not clear whether representing text in terms of syllables will yield performance improvement in E2E speech recognition.

However, syllable-based ASR schemes have never been tried with E2E models in Indian languages. The earlier works on syllable-based tokens use conventional Gaussian mixture model (GMM)-hidden Markov model (HMM) systems (Lakshmi & Murthy, 2006; Panda & Nayak, 2016). The suitability of syllables as a token unit for text representation in E2E ASR systems for Indian languages is explored in this work. Specifically, this work attempts to answer the following queries.

1. Can syllables be practical alternatives for graphemes as modeling units for E2E ASR in a monolingual training setup?
2. Can syllable-based tokens supplement the grapheme-based tokens through

a pretraining-finetuning setup, such that the model's performance improves?

3. How good are syllable-based models in cross-lingual transfer learning? When pretrained on high-resource languages, can they transfer well to a related low-resource language?

In an attempt to address the questions above, the text sequence is encoded with three different units: grapheme units from the native script, units from the Sanskrit library phonetic (SLP1) encoding scheme (Scharf, 2013) and syllables. SLP1 is an ASCII transliteration scheme from and to the native script. It maps both the vowel and the vowel modifier graphemes to the same ASCII character. Representations based on syllables use SLP1 to generate the syllable sequence. Each syllable is mapped to a single Unicode character for text representation. Three types of sub-word modeling units are explored for acoustic and language modeling in ASR: character, byte pair encoding (BPE) (Sennrich et al., 2016), and unigram language modeling (ULM) (Kudo, 2018). The performance of syllable-based units is compared with those of other tokens in the ASR task for several Indian languages.

The novelties of this work are as follows:

1. To the best knowledge of the authors, this is the maiden work on end-to-end speech recognition in Indian languages with syllables as the token unit.
2. This is one of the first works validating the performance of syllable-based tokens in cross-lingual transfer learning for Indian languages.

2. Related Works

E2E ASR has become quite popular over the last few years since it can directly map the input acoustic speech signal to the grapheme or word sequence. Transformer architecture (Vaswani et al., 2017), originally proposed for machine translation task, was a milestone in E2E sequence to sequence (*seq2seq*) models. It gets rid of the recurrence in *seq2seq* (Prabhavalkar et al., 2017; Cho et al.,

2014) models and allows more parallelization. The main component of the transformer architecture is a self-attention module that tries to capture the dependencies between different positions through a position-pair computation. In RNNs, this computation is sequential (one step at a time) whereas it is one-time in transformers. The transformer architecture was successfully adapted to speech recognition in (Dong et al., 2018; Karita et al., 2019b) and Karita et al. (2019a) improved the transformer-based speech recognition with joint connectionist temporal classification (CTC) (Graves et al., 2006) training and integration of a language model (LM) during decoding.

Though transformers are good at modeling long-range global context, they are less effective in capturing the local feature patterns. On the other hand, convolutional neural networks (CNN) are good at capturing the local patterns but need more layers to capture the global context. Gulati et al. (2020) proposed a conformer encoder that judiciously combines self-attention and convolutions to reap the benefits of both. A conformer block has four modules stacked together: i) a feed-forward module, ii) a self-attention module, iii) a convolution module, and iv) a second feed-forward module in the end. This architecture achieves state-of-the-art accuracies across multiple speech datasets (Guo et al., 2021).

Languages in the Indian subcontinent have a strong correspondence between the graphemes in the native script and their pronunciation (sound units). Hence graphemes in the native script are used as the modeling units for building end-to-end ASR in Indian languages. Also, the phoneme sets of most Indian languages are almost the same. Many of the multilingual models try to leverage this commonness by transliterating the text to a common character/phoneme set (Shetty & Umesh, 2021; Kumar et al., 2021). Such transliterations are also used to eliminate the rendering errors in code-switched context (Diwan et al., 2021; Khare et al., 2021; Datta et al., 2020; Thomas et al., 2020) and to improve the monolingual ASR performance (Adiga et al., 2021; Anoop & Ramakrishnan, 2021b). Common label set (Shetty & Umesh, 2021; Prakash et al., 2019), SLP1 (Adiga et al., 2021; Anoop & Ramakrishnan, 2021b, 2022), and Devanagari script are some of the schemes usually employed to provide shared labeling

schemes. Similar sounding letters in different Indian languages occur at the same offset from the starting of the allocated range for each script in the Unicode tables. This aspect makes such mapping schemes simple in the Indian context.

Syllable-based schemes are employed in E2E frameworks for Mandarin Chinese ASR by Zhou et al. (2018) and Qu et al. (2017). Both works report that syllable-based models perform better than context-independent (CI) phonemes. Zhou et al. (2018) use transformer models while Qu et al. (2017) use long-short-term memory (LSTM) RNNs trained with CTC and state-level minimum Bayes risk (sMBR) loss. Zhao et al. (2019) use syllables in a multitask recognition framework to achieve simultaneous speech content recognition, dialect identification, and speaker recognition in Tibetan language.

Subword modeling for E2E ASR has recently become popular with the introduction of subword tokenization algorithms like byte-pair-encoding (Sennrich et al., 2016) and unigram language model (Kudo, 2018). The core idea of both BPE and ULM is to encode the text with some data compression scheme. BPE uses a dictionary encoder that incrementally finds a set of symbols, such that the total number of symbols encoding the text is minimized. BPE first splits the whole utterance into individual grapheme units. The most frequent adjacent pairs of grapheme units are successively merged till the desired vocabulary size is reached. Unlike BPE, ULM considers multiple segmentations during vocabulary creation with a probabilistic unigram language model. The probability of each of these subword sequences is computed as the product of the unigram probabilities of the constituent subwords. Starting from a seed vocabulary, it iteratively finds the most valuable subwords (which cause the maximum reduction in the likelihood of data when removed) till the desired vocabulary size is reached. The seed vocabulary is usually chosen as the union of all the grapheme units and their most frequent substrings in the corpus. Subword units have been used for speech recognition in Drexler & Glass (2019); Rao et al. (2017); Lakomkin et al. (2020) and Xiao et al. (2018).

3. Experimental Setup

This section details the experimental setup, specifically the datasets used in the experiments, the extraction of acoustic features, the linguistic modeling units used, the architectures for the acoustic and language modeling, and the decoding scheme.

3.1. Datasets

Preliminary experiments are conducted for Sanskrit and then repeated for several other Indian languages. *Vāksañcayaḥ* dataset (Adiga et al., 2021) is used for end-to-end speech recognition in Sanskrit as it is the largest publicly available ASR dataset in Sanskrit. The only other public dataset in Sanskrit, *Shrutilipi* (Bhogale et al., 2022), has only 27 hours of annotated speech. Also, *Shrutilipi* data is collected from only the radio news bulletins and hence lacks diversity. *Vāksañcayaḥ* has around 79 hours of data with a sampling rate of 22050 Hz. The data is also diverse and includes the readings of various texts in Sanskrit literature, contemporary stories, radio programs, extempore discourse, etc. The dataset has four subsets: train, validation, test, and out-of-domain (OOD) test. The approximate ratio of this split is 71:9:14:6. The details of these subsets are summarised in Table 1. The speakers of the different subsets are disjoint. The OOD test set has content that largely differs from the domain of the training set or has speakers with a pronounced influence of their native language.

The experiments are repeated for the best combinations of graphemes and subword units in three other languages - Telugu, Tamil, and Odia using the MUltilingual and Code-Switching (MUCS-2021) dataset (Diwan et al., 2021; [dataset] Telugu and Tamil, 2018). The effectiveness of syllable-based modeling units in complementing grapheme-based models are explored with these datasets. The usefulness of top-performing representations in a transfer learning setup is explored next. Here the models are trained using Sanskrit and Telugu data and finetuned in a related language, Kannada. The OpenSLR dataset

Dataset	Data split	Train	Validation	Test	OOD
Sanskrit *Vāksañcayaḥ*	Hours	56	7	11	5
	Utterances	34309	3337	5529	2778
	#Unique	32324	3271	5524	2776
Telugu *MUCS 2021*	Hours	40	5	4	-
	Utterances	44882	3040	2549	-
	#Unique	34176	2997	2506	-
Tamil *MUCS 2021*	Hours	40	5	4	-
	Utterances	39131	3081	2609	-
	#Unique	30205	3055	2584	-
Odia *MUCS 2021*	Hours	94.5	5	5.5	-
	Utterances	59782	3471	4420	-
	#Unique	820	65	124	-
Kannada† *SLR79*	Hours	-	4	4	-
	Utterances	-	2000	2223	-
	#Unique	-	1702	1881	-

† Note: Sanskrit + Telugu speech data (72 hours, 67979 utterances, 55300 unique sentences) is used for pretraining the Kannada model. Utterances having syllables not present in the Kannada text corpus are removed from the training set.

Table 1: Details of the ASR speech datasets used in this work. The Kannada dataset is used only for cross-lingual transfer learning.

SLR79 (He et al., 2020) is used for Kannada. After removing the sentences with English alphabets, the remaining data is split into 4 hours each to form the test and validation subsets. The utterances with syllables not present in the Kannada text corpus are filtered out, resulting in 72 hours of speech data for pretraining. The validation set used during the pretraining stage is also used for finetuning the Kannada models, as the available data in Kannada is limited.

The text corpora used for language modeling in Sanskrit, Telugu, Tamil, Odia, and Kannada have 0.28, 0.9, 1.3, 0.1, and 0.6 million sentences, respectively. The text data is collected from *wiki* data dump ([dataset] Wiki text data

dump, 2020).

3.2. Feature preparation

80-dimensional mel filterbank features computed with 25 ms windows and 10 ms strides are used as the acoustic features in the experiments. The details of the feature extraction process are shown in Figure 1. Cepstral mean variance normalisation (CMVN) is applied at the utterance level. Speed perturbation (Ko et al., 2015) with factors 0.9, 1.0, and 1.1 are used for robust ASR training. Spectrogram augmentation (Park et al., 2019) is also applied with time warping, frequency masking, and time masking parameters of 5, 30, and 40, respectively.

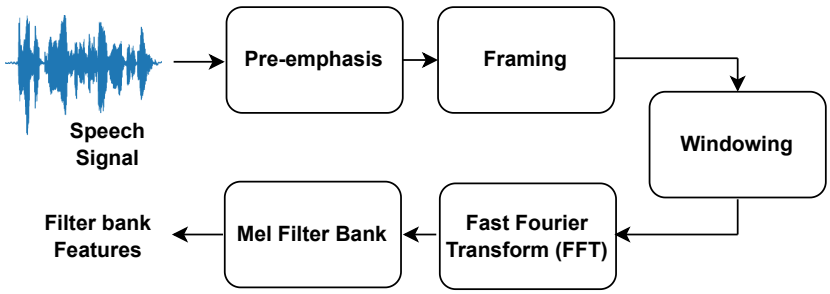

Figure 1: Extraction of mel filterbank features

3.3. Different linguistic units used in the study

This work represents text in three different ways: native script, SLP1 encoding scheme (Scharf, 2013) and syllable-based representation. The native script for Sanskrit is Devanagari. There are a total of 78 distinct character units in Devanagari, including vowels and their modifiers like ा(आ), ि(इ), and ी(ई). The script uses vowel modifiers when a vowel combines with a consonant. These vowel modifiers cause redundancy in text representation since multiple grapheme units are possible for the same phonemic unit.

SLP1 mapping scheme is available at (SLP1 mapping, 2015). There are 54 distinct SLP1 units in Devanagari.

Type	Text
Native	इदानीम् विचारणा काचित् प्रचलति
SLP1	idAnIm vicAraRA kAcit pracalati
Syllable	i-dA-nIm vi-cA-ra-Ra kA-cit pra-ca-la-ti

Table 2: Different representations for Sanskrit text. Devanagari is the native character set for Sanskrit. In syllable representation a unique unicode character is assigned for each distinct syllable so as to facilitate the discovery of sub-word units using `sentencepiece`.

In the third case, the SLP1 encoded text is represented as a sequence of syllables, with each syllable represented by a unique Unicode character. First, a list of syllables in the language is created from the text corpus. Each of the syllables is then encoded with a unique Unicode character. This makes the syllable the basic grapheme unit for ASR. As in the `Transliterate` tool from `https://sanskritlibrary.org/`, a simple syllable conversion scheme is used in the experiments. Table 2 shows the text representations using the different schemes for a sample Sanskrit sentence.

These text representations are used to build sub-word units for acoustic and language modeling in ASR. Specifically, BPE (Sennrich et al., 2016) and ULM (Kudo, 2018) are used for building the sub-word units. `sentencepiece`, a Python module is used for training the tokenization and detokenization models from the raw sentences.

3.4. Acoustic modeling

Figure 2 shows the network architecture used for acoustic modeling. Conformer blocks (Gulati et al., 2020) are used in the encoder, and transformer blocks (Vaswani et al., 2017) are used in the decoder. Interested readers may follow the above references for more details on the structures of conformer encoder and transformer decoder blocks. Table 3 lists the chosen values of the model hyperparameters. Slightly higher size models with attention dimension (d^{att}) of 512 are used for the monolingual training in Sanskrit and the cross-lingual training in Kannada. More than 50 hours of training data are available

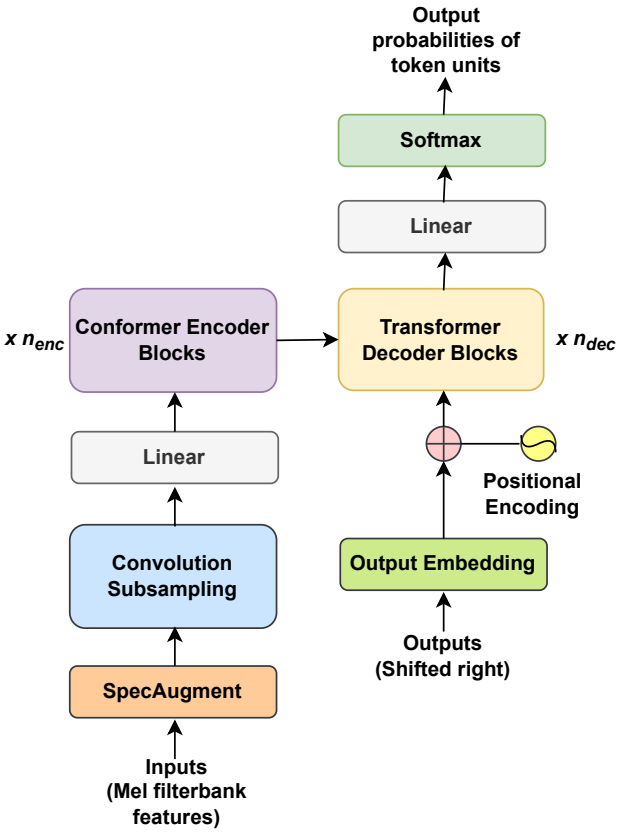

Figure 2: The end-to-end architecture used for acoustic modeling in this work.

in these cases from diverse domains. Attention dimensions d^{att} of 256 are used for the other languages. Choosing a smaller configuration helps us accelerate experimentation with faster training and lesser inference time.

CTC and attention weights of 0.3 and 0.7 are used during training. A 12-layer encoder network and a 6-layer decoder network are used, each with 2048 units, with a 0.1 dropout rate. Each layer contains eight 64-dimensional attention heads, which are concatenated to form a 512-dimensional attention vector. The convolutional subsampling uses a two-layer CNN with 256 channels, stride 2, and kernel size 3. The learning rate coefficient is ten, and the CONV module kernel size is 31. Noam optimizer (Vaswani et al., 2017) is used for training with a learning rate of 10 and warmup steps of 25000. Training is performed for a

Model	DataAugment		$kernel$	n_{enc}	n_{dec}	H	d_{ff}	d_{att}
	Speed Perturb	Spec Aug						
Sanskrit Kannada	✓	✓	31	12	6	8	2048	512
Telugu Tamil Odia	✓	✓	15	12	6	4	2048	256

Table 3: Values of the hyperparameters of the ASR models used for each dataset. The Table lists the data augmentation techniques, kernel size ($kernel$) of the conformer encoder, number of encoder (n_{enc}) and decoder layers (n_{dec}), number of attention heads (H), feed-forward layer dimension (d_{ff}) and attention dimension (d_{att})

Embedding dimension	128
No. of encoder layers	16
No. of attention heads, H	4
Attention dimension, d^{att}	512
FF layer dimension, d^{ff}	2048

Table 4: Hyperparameters used while training the transformer-based language models.

maximum of 50 epochs with a patience value of 10 for early stopping. The top 3 models with the best validation accuracy are averaged and used for decoding.

3.5. Language modeling

Transformer architecture is used for language modeling (LM). The hyperparameters used in the architecture are given in Table 4. The model is trained for a maximum of 20 epochs with patience of 6. Adam optimiser is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of 0.0001. All the experiments are performed using the ESPNet toolkit (Watanabe et al., 2018).

3.6. Decoding

Hybrid CTC-attention scheme (Watanabe et al., 2017b) is used for decoding. Given the speech input filterbank features X, the most probable token sequence $\hat{C}$ is obtained as

$$\hat{C} = \underset{C \in V^*}{\arg\max}\, \lambda\ log\ p_{ctc}(C|X) + (1-\lambda) log\ p_{att}(C|X) + \gamma\ log\ p_{lm}(C) \qquad (1)$$

p_{ctc}, p_{att}, and p_{lm} are the CTC, attention, and language model scores, and V^* is the set of all target hypotheses. The hyperparameters during decoding are: beam size = 10, CTC weight (λ) = 0.5, and language model weight (γ) = 0.6.

4. Results and Discussion

Results of the ASR experiments with different types of tokens are summarized in Table 5. The syllable and the sub-word-based tokens increase the vocabulary size and reduce the length of the text representations. So, instead of the token error rate (TER), the character error rate (CER) is reported allowing a fair comparison between the different representations. The results in bold are the best values of CER and word error rate (WER) with the different tokens employed in this work. Syllable-BPE representation performs the best among all the different text representations for the Sanskrit dataset. Results with BPE and ULM sub-word tokens are close in the case of syllable representations. Grapheme representation with the native script performs the worst with all the sub-word units - character, BPE, and ULM.

Among character-level tokens, the use of SLP1 performs better. The results with the native script are poorer than the results with SLP1. A major reason for the poor performance of the native script is the confusion as to whether to output the vowel or its modifier. Errors due to the absence of virama (◌्, also called halant) are also common. Repetitions of vowels/characters and the confusion between short (unstressed) and long (stressed) versions of vowels are more frequent with the native script than SLP1. Several such examples are shown in Table 6.

			Test		OOD	
Vocab Type	Grapheme Units	Vocab Size	CER	WER	CER	WER
Char	Native	78	14.7	74.8	19.7	82.8
	SLP1	54	**2.5**	**14.1**	**6.7**	**33.5**
	Syllable	14416	5.3	28.5	12.1	50.5
BPE	Native	2000	15.4	41.6	20.8	54.6
	SLP1	2000	5.2	17.2	10.7	34.7
	Syllable	16000	**2.2**	**8.1**	**8.0**	**26.4**
ULM	Native	2000	14.6	41.8	20.4	55.5
	SLP1	2000	7.3	23.0	13.2	39.5
	Syllable	16000	**2.3**	**8.2**	**8.1**	**27.2**

Table 5: Recognition results on the test and OOD sets of Vāksañcayaḥ dataset. There are 35266 words in the test set and 21672 words in the OOD set.

Another observation is that the sandhi rules in Indian languages make the WER values look worse than it actually is. Speakers may split a word or merge two words at their boundary, which information is not reflected in the transcription. Such issues inflate the WER by a significant factor. Two such examples are listed in Table 7. In the first case, the decoded text is perfect. However, WER is penalized with four errors. In the second case, all the seven errors are due to incorrect word boundaries and long/short vowel confusions in the prediction.

4.1. Additional studies

4.1.1. Impact of language models on the ASR performance

The results with and without the LM are compared for the best combinations of grapheme and sub-word units indicated by Table 5 to find out the contribution of the language model (LM). The results are shown in Table 8. There is a substantial boost in performance (reduction in WER) with language

Type	Native Text representation	SLP1 Text representation
REF	तदा ब्रह्मा **तान्** इत्थम् **उवाच** इयम् भूः विष्णोः पत्नी	tadA brahmA **tAn** itTam **uvAca** iyam BUH vizRoH patnI
HYP	तदा ब्रम्म **तआन्** इत्थम् **उवआच** इयम् भूः विष्णोः पत्नी	tadA brahma **tAn** itTam **uvAca** iyam BUH vizRoH patnI
REF	**किमर्थम्** एवम् इति	**kimarTam** evam iti
HYP	**किमरथम्** एवम् इति (halant missed)	**kimarTam** evam iti
REF	पञ्च वर्षाणि **अतीतानि**	paYca varzARi **atItAni**
HYP	पञ्च वर्षऽणि **अआतीतानि** (vowel repeated)	paYca varzARi **atItAni**

Table 6: Common rendering errors in decoding with native script, that get corrected with SLP1 representation. REF and HYP refer to the reference and decoded sequences, respectively.

Type	SLP1 Text
REF	**** ahantu nimitta mAtramevAsmi
HYP	aham tu nimittamAtrameva asmi
Eval	I S S S
REF	ezA ca nadI prAkpaScimadiSayoH pravahantI Buvam ***** sasyaSyAmalAm kurvatI virAjate
HYP	*** ezAcanadi prAk paScimadiSayoH pravahanti Buvam sasya SyAmalAm kurvatI virAjate
Eval	D S S S S I S

Table 7: Examples where sandhi rules make the WER worse than it actually is. D, I, and S indicate deletion, insertion, and substitution errors, respectively.

models. For the test set, the reduction in WER for SLP1-Char, Syllable-BPE, and Syllable-ULM tokens are 13.5%, 21.1%, and 19.1%, respectively, with the incorporation of LM. The corresponding reductions in the WER on the OOD set are 16.9%, 21.8%, and 19.5%, respectively. In both the datasets, maximum improvement in performance is obtained with syllable BPE representation, irrespective of whether it is measured in terms of CER, WER or SER.

4.1.2. Comparison with previous work

The experimental results are compared with the previous work (Adiga et al., 2021) on the *Vāksañcayaḥ* dataset. Their acoustic model uses time delay neural networks (TDNN) (Peddinti et al., 2015). They use n-gram language models with Kneser-Ney smoothing. They use 40-dimensional mel frequency cepstral coefficients (MFCC) along with 100-dimensional i-vector-based speaker embedding (Saon et al., 2013). Their best results are with BPE as the unit of language model and grapheme as the unit of acoustic model; both represented using SLP1.

Particulars			Test			OOD		
Vocab Type	Vocb Size	LM	CER	WER	SER	CER	WER	SER
SLP1-Char	54	✗	4.5	27.6	73.3	10.2	50.4	93.2
SLP1-Char	54	✓	2.5	14.1	43.9	6.7	33.5	76.5
Syllable-BPE	16000	✗	5.8	29.2	75.5	12.1	48.2	90.6
Syllable-BPE	16000	✓	2.2	8.1	25.8	8.0	26.4	57.3
Syllable-ULM	16000	✗	5.3	27.3	73.3	11.2	46.7	89.9
Syllable-ULM	16000	✓	2.3	8.2	26.6	8.1	27.2	59.5

Table 8: Performance of syllable-based sub-word units on Sanskrit Vāksañcayaḥ test and OOD sets with and without the use of language models (LM). SER stands for sentence error rate.

Table 9 compares the results from the conformer architecture trained using syllable BPE units with their results. Syllable BPE tokens give 17.3% improvement (WER reduction) on OOD set and 13.8% improvement on the test set. The results consistently improve on each of the subdivisions of the OOD set.

No.	Dataset description	TDNN+SLP1 (Adiga et al., 2021)	Conformer + Syllable-BPE
I.	Average on OOD dataset	43.7†	26.4
a	*Tamil influenced accents*	*34.9*	*14.0*
b	*Hindi influenced accents*	*39.0*	*11.5*
c	*Radio program*	*46.3*	*31.7*
d	*Extempore discourse*	*48.9*	*39.3*
e	*Live lecture*	*47.9*	*33.8*
II.	Average on Test dataset	21.9	8.1

† The average on OOD set is computed from the results reported on individual subsets.

Table 9: Comparison of the results (WER in %) with those of the earlier work on Vāksañcayaḥ OOD and test sets.

4.1.3. Effect of vocabulary size

The performance of the ASR system is evaluated with different vocabulary sizes for BPE and ULM tokens. The results are shown in Figure 3. It is evident that increasing the vocabulary size does not help improve the WER. The best performance is when the BPE/ULM vocabulary size is kept closer to the number of character tokens.

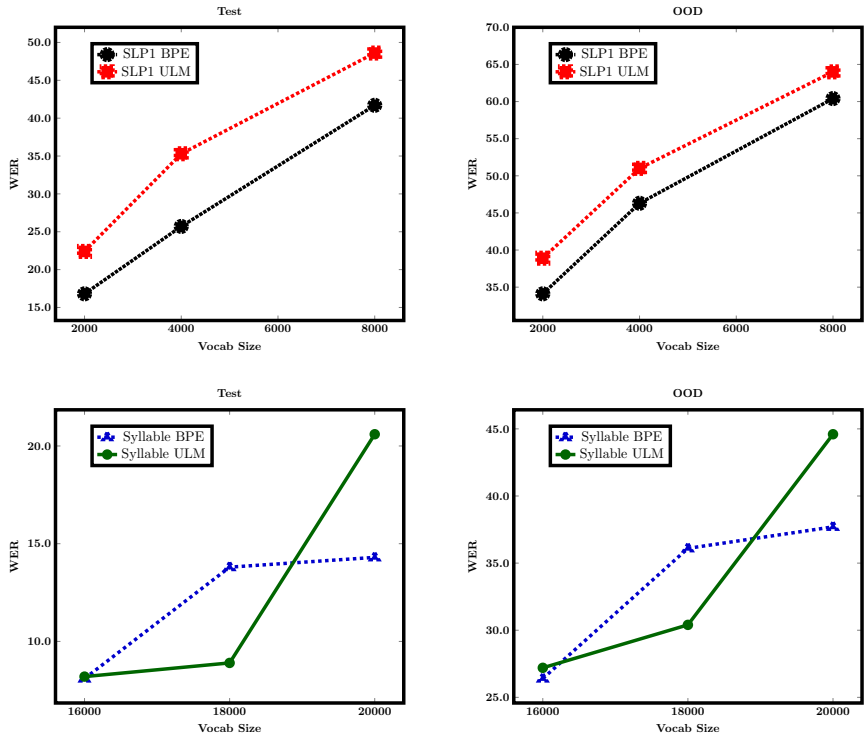

Figure 3: Effect of vocabulary size on the ASR performance (in terms of % WER) on the test and OOD test data of *Vāksañcayaḥ* dataset for SLP1 and syllable BPE/ULM tokens.

4.1.4. Performance of syllable BPE on other Indian languages

To verify whether such an improvement with syllable BPE units translates to other Indian languages, the performances of these textual representations on Telugu, Tamil, and Odia data of the MUCS-2021 dataset are evaluated. Since the current study focuses on the effectiveness of different representations for

text, only three languages in the MUCS-2021, where the schwa deletions are absent and the mapping of the native script to SLP1 is quite direct, are considered. However, SLP1 mapping can also be extended to languages with schwa deletion property, with an additional step to remove the schwa based on context. Though the Odia train set has a total of 59782 utterances, it has only 820 unique sentences (Refer Table 1). The choice of the Odia dataset was particularly motivated by this aspect. The dataset allows verifying the performance of syllable-BPE in a stringent setting. The experiments are conducted with an attention dimension of 256 and attention heads of four (a smaller model) in these languages. Since syllable-based BPE and ULM tokens give almost similar results, only SLP1-char and syllable-BPE tokens are used in these experiments. All the models are trained for 50 epochs. The results are shown in Table 10.

Dataset	#Unique sentences	Vocab type	Vocab size	CER	WER	WER MUCS Baseline	WER MUCS CSTR
Telugu	34176	SLP1-Char	57	7.8	30.0	31.4	19.7
		Syllable-BPE	16000	7.2	**26.0**		
Tamil	30329	SLP1-Char	40	6.5	28.4	34.1	21.7
		Syllable-BPE	15000	6.9	**26.1**		
Odia	820	SLP1-Char	52	13.0	**40.2**	38.5	25.4
		Syllable-BPE	8000	71.6	91.4		

Table 10: Results on the Telugu, Tamil, and Odia datasets. The penultimate column lists the MUCS-2021 baseline WER (Diwan et al., 2021) and the last column lists the WER of the CSTR system (Klejch et al., 2021) that won the MUCS-2021 challenge. They employ multilingual pretraining and monolingual finetuning. CSTR system uses additional Youtube data for semi-supervised training.

Syllable-BPE tokens give the best WER performance on the Telugu and Tamil datasets, outplaying the SLP1-Char by 4% and 2.3%, respectively. Both the tokens yield poor results on the Odia dataset, and syllable BPE tokens yield abysmal results compared to the SLP1-Char. Since the Odia train set

has only 820 unique sentences, the model overfits during syllable-BPE training. Overfitting is an undesirable behavior in machine learning models where they give accurate predictions on the training data but perform poorly on the unseen data. Models should generalize well, giving accurate predictions across all types of data within their domain. But overfitting causes the model to fit too closely to the training dataset and limits its generalization capability. So they perform well on the training data but fail on the test data. The major reasons for overfitting are the small size of the training dataset and the lack of diversity in the data. The Odia training dataset does not contain enough samples to represent the possible input syllable sequences. Due to lack of diversity in the data, the model overfits the training syllable sequences, and the training accuracy nears the 100% mark in less than five epochs. But performance on the test data is poor as the model lacks the generalization capability. The results suggest the necessity for diversity in the training data while training syllable-based models.

Table 10 also lists the baseline, and the best WER (CSTR system (Klejch et al., 2021)) reported on the MUCS-2021 challenge. Both of them use sequence-discriminative training with lattice-free maximum mutual information (LF-MMI) (Povey et al., 2016) objective. They employ TDNN architecture and an explicit lexicon in a multilingual training setup. Syllable-BPE tokens beat the MUCS baseline WER for Telugu and Tamil by 5.4% and 8%, respectively. However, the obtained results are 6.3% absolute worse in Telugu and 4.4% absolute worse in Tamil than the CSTR system - the challenge winners. CSTR system employs multilingual training with a long ASR pipeline and uses more training data than the system employed in this work. They train a GMM first with an explicit pronunciation lexicon. The alignments obtained from the GMM model are used to train a CNN-TDNN acoustic model in a multilingual fashion. This step pools data from multiple languages and uses language-specific output layers. This multilingual model is then finetuned on monolingual data to generate language-specific seed models. They collect around 200 hours of additional data for each language from Youtube which is then decoded with the seed models. The decoded transcriptions are used for semi-supervised mul-

tilingual training followed by monolingual finetuning. They also use recurrent neural networks to rescore the lattices obtained in the first pass decoding with trigram language models. In contrast, the experiments in the current study use a monolingual setup since the focus is on comparing the performance of different text tokens in end-to-end transformer-based models. Thus, apart from the data provided with the challenge, no additional data is used in the experiments. Also, no explicit lexicon is used. The training of acoustic and language models involves just a single stage in the current study.

4.1.5. Testing if syllable-BPE and SLP1-char models supplement each other in a pretraining-finetuning setup

The network is first pretrained with one token type and then finetuned with the other. One model uses SLP1-Char for pretraining and syllable-BPE for finetuning (SLP1-Char → Syl-BPE). The other uses syllable-BPE for pretraining and SLP1-Char for finetuning (Syl-BPE → SLP1-Char). The pretrained and finetuned models have the same architecture except for the final linear layer. The entire network for finetuning is initialized with weights from the pretrained model. All the layers are finetuned for 50 epochs. The experiments are performed on Sanskrit, Telugu, and Tamil datasets. The results are listed in Table 11.

Dataset	CER/WER			
	SLP1-Char	Syl-BPE	SLP1-Char → Syl-BPE	Syl-BPE → SLP1-Char
Sanskrit/Test	2.5/14.1	2.2/**8.1**	2.6/11.2	**1.5**/9.3
Sanskrit/OOD	6.7/33.5	8.0/**26.4**	7.2/29.4	**5.1**/27.9
Telugu/Test	7.8/30.0	7.2/26.0	**7.1**/**25.8**	7.6/28.7
Tamil/Test	**6.5**/28.4	6.9/26.1	**6.5**/**25.8**	**6.5**/28.5

Table 11: Performance of models pretrained with one type of token and finetuned on the other.

CER improves over the models directly trained on SLP1-Char (second col-

umn) when pretrained with Syl-BPE and finetuned on SLP1-Char (fifth column). On the Sanskrit test and OOD sets, this scheme gives CER improvements of 1% and 1.6%, respectively, over the model directly trained on SLP1-Char. On the Telugu test set, CER improves by 0.2%, whereas there is no improvement on the Tamil test set. The corresponding WER improvements (columns 2 and 5) on the Sanskrit-test, Sanskrit-OOD and Telugu-test sets are 4.8%, 5.6%, and 1.3%, respectively. However, these improved WERs still do not match the performance of Syl-BPE trained models. Once again, there is no improvement in WER observed for Tamil.

Models pretrained with SLP1-Char and finetuned on Syl-BPE (fourth column) give marginal improvements in CER and WER over the models trained only on Syl-BPE (third column) for Telugu and Tamil datasets. CER improves by 0.1% for Telugu and 0.4% for Tamil. The corresponding WER improvements are 0.2% for Telugu and 0.3% for Tamil. However, no WER benefits are observed for Sanskrit.

Pretraining-finetuning setup makes the training a two-stage process, increasing the training time and cost. However, the performance improvement in WER (if any) over the models trained with syllable BPE is minor. Hence syllable BPE seems to be a reasonable choice for training the end-to-end systems.

4.1.6. Exploring the effectiveness of syllable-based units for transfer learning among related languages

Syllable-based sub-word units seem to work well in a monolingual setup (trained and tested on the same language), provided the dataset is diverse enough. The conformer models are able to capture the inter-relationships between the different syllable tokens in an utterance through the self-attention mechanism. Many Indian languages share common root verbs and nouns. Hence, the association between the syllables learned with a model trained in one language may provide a useful initialization for another. However, the languages differ in their grammar and the use of inflections at the end of the words. In this context, it is unclear whether using syllable-based tokens for

cross-lingual transfer (pretraining with one or more languages and finetuning with a related target language on which the model is tested) can help improve the ASR performance of the target language. It differs from monolingual training in that it starts with a model pre-initialized with data from other languages instead of random initialization. To answer the above question, a conformer system is trained using speech data from Sanskrit (train, dev, test, and OOD sets) and Telugu (train and test sets) and tested on the Kannada speech corpus. The text corpus for training the LM contains approximately 0.6 million utterances collected from the wiki text data dump for Kannada. Syllable-BPEs are trained from the Kannada text corpus. Only those utterances with their syllables present in the LM text corpus are considered for AM training. This amounts to around 68 K utterances. The performance of the conformer model with SLP1-Char and Syllable-BPE tokens are shown in Table 12. The acoustic model is trained for 30 epochs with a patience value of 10.

Vocab type	Vocab size	Zero-shot		finetune	
		CER	WER	CER	WER
SLP1-Char	55	11.3	56.7	2.7	18.9
Syllable-BPE	25000	31.3	95.7	-	-

Table 12: Results on the Kannada test set with a model pretrained on Sanskrit + Telugu and finetuned on Kannada.

Even after 30 epochs of training with syllable-BPE tokens, the accuracy on the Kannada validation set using greedy decoding fails to reach the 30% mark. On the other hand, greedy decoding yields around 75% accuracy on the validation set with SLP1-Char models trained for 30 epochs. Hence the zero-shot accuracies with SLP1-Char on the test set are much better than the syllable-BPE units. Finetuning with 4 hours of Kannada speech data for 30 epochs improved the SLP1-Char WER on the test set from 56.7% to 18.9%. Finetuning the syllable-BPE model is quite unstable since the pretrained model underfits, and the amount of finetune data is quite limited. This suggests that the

SLP1-Char units are much better than the syllable-BPE ones for cross-lingual transfer learning. The failure of syllable-BPE in this case maybe attributed to the following reasons. Firstly, different Indian languages have different syllable probabilities. Their co-occurrence with other syllables also will be different in different languages. For example, the visarga (◌ः) is common in Sanskrit but rarely used in other languages. Similarly, the usage of డ(da) is more common in Telugu. Secondly, the rules for inflection (vibhakthi) are different in different Indian languages, which affect the association of individual syllables with others. Hence, the self-attention-based conformer models might not work well with syllable-based units in a cross-lingual transfer learning setup with a limited amount of target speech data.

4.2. Limitations of using syllable-based tokens

Syllables reduce the effective length of token sequences compared to using characters. Hence they better capture the associations between tokens than characters in a monolingual setup. However, there are some limitations as well. Using syllables increases the vocabulary size, thereby increasing the size of the softmax layer at the network's output. This increases the number of learnable parameters in the model. Also, syllable-based tokens do not help in cross-lingual transfer learning, since different languages have different syllable distributions and different associations between the syllables.

5. Conclusions

Syllable-based sub-word units seem to be a promising alternative to grapheme-based schemes in the monolingual training setup if the dataset is diverse enough to ensure fair coverage of the syllables in the language. Syllable-BPE tokens provide state-of-the-art results on the Vāksañcayaḥ dataset in Sanskrit. The language models are more significant during the decoding of syllable-based tokens than the grapheme-based tokens. However, syllable-based schemes fail when the training data is not diverse enough, like in the case of MUCS -2021

Odia dataset. Performance improvements in WER (if any) with a pretraining-finetuning setup involving syllable-BPE and SLP1-char tokens are minor compared to the syllable-BPE trained model. Also, the syllable-based units do not seem suitable for cross-lingual transfer learning to related Indian languages, since different languages have different distributions of syllables.

6. Future work

Syllable BPE units exhibit encouraging results for monolingual speech recognition. However, they are not suitable for cross-lingual transfer learning, and SLP1 character units perform better than them. Thus there is a motivation to study the performance of SLP1 character-based tokens in a multilingual speech recognition environment for Indian languages in the future. Also, it would be interesting to see if using both token units in a multitask learning framework can help improve ASR performance.

7. Acknowledgment

The authors thank Science and Engineering Research Board, Government of India for partially funding this research through the IMPRINT2 project, IMP/2018/000504.

References

Adiga, D., Kumar, R., Krishna, A., Jyothi, P., Ramakrishnan, G., & Goyal, P. (2021). Automatic speech recognition in Sanskrit: A new speech corpus and modelling insights. *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, (pp. 5039–5050).

Anoop, C. S., & Ramakrishnan, A. G. (2021a). CTC-based end-to-end ASR for the low resource Sanskrit language with spectrogram augmentation. In *Proc. of the National Conference on Communications (NCC)* (pp. 1–6).

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

Anoop, C. S., & Ramakrishnan, A. G. (2022). Exploring a unified ASR for multiple South Indian languages leveraging multilingual acoustic and language models. In *Proc. of the IEEE Spoken Language Technology Workshop (SLT)* (pp. 830–837).

Bhogale, K. S., Raman, A., Javed, T., Doddapaneni, S., Kunchukuttan, A., Kumar, P., & Khapra, M. M. (2022). Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages. *arXiv*, *abs/2208.12666*.

Chan, W., & Lane, I. (2016). On online attention-based speech recognition and joint Mandarin character-Pinyin training. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 3404–3408).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734).

[dataset] Telugu and Tamil (2018). Data provided by Microsoft and SpeechOcean.com. URL: `https://msropendata.com/datasets/7230b4b1-912d-400e-be58-f84e0512985e`.

[dataset] Wiki text data dump (2020). Wikipedia dumps. URL: `https://dumps.wikimedia.org/`.

Datta, A., Ramabhadran, B., Emond, J., Kannan, A., & Roark, B. (2020). Language-agnostic multilingual modeling. In *Proc. of the IEEE Interna-*

tional Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8239–8243).

Diwan, A., Vaideeswaran, R., Shah, S., Singh, A., Raghavan, S., Khare, S., Unni, V., Vyas, S., Rajpuria, A., Yarra, C., Mittal, A., Ghosh, P., Jyothi, P., Bali, K., Seshadri, V., Sitaram, S., Bharadwaj, S., Nanavati, J., Nanavati, R., & Sankaranarayanan, K. (2021). MUCS 2021: Multilingual and code-switching ASR challenges for low resource Indian languages. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 351–355). volume 1.

Dong, L., Xu, S., & Xu, B. (2018). Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5884–5888).

Drexler, J., & Glass, J. (2019). Subword regularization and beam search decoding for end-to-end automatic speech recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6266–6270).

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. of the ACM International Conference* (pp. 369–376). volume 148.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented transformer for speech recognition. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 5036–5040).

Guo, P., Boyer, F., Chang, X., Hayashi, T., Higuchi, Y., Inaguma, H., Kamo, N., Li, C., Garcia-Romero, D., Shi, J., Shi, J., Watanabe, S., Wei, K.,

Zhang, W., & Zhang, Y. (2021). Recent developments on ESPNet toolkit boosted by conformer. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5874–5878).

He, F., Chu, S.-H. C., Kjartansson, O., Rivera, C., Katanova, A., Gutkin, A., Demirsahin, I., Johny, C., Jansche, M., Sarin, S., & Pipatsrisawat, K. (2020). Open-source multi-speaker speech corpora for building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu speech synthesis systems. In *Proc. of the 12th Language Resources and Evaluation Conference (LREC)* (pp. 6494–6503). European Language Resources Association (ELRA).

Karita, S., Kubo, Y., Bacchiani, M., & Jones, L. (2021). A comparative study on neural architectures and training methods for Japanese speech recognition. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech*.

Karita, S., Soplin, N., Watanabe, S., Delcroix, M., Ogawa, A., & Nakatani, T. (2019a). Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 1408–1412).

Karita, S., Wang, X., Watanabe, S., Yoshimura, T., Zhang, W., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N., & Yamamoto, R. (2019b). A comparative study on transformer vs RNN in speech applications. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 449–456).

Khare, S., Mittal, A., Diwan, A., Sarawagi, S., Jyothi, P., & Bharadwaj, S. (2021). Low resource ASR: The surprising effectiveness of high resource transliteration. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 1051–1055). volume 2.

Klejch, O., Wallington, E., & Bell, P. (2021). The CSTR system for multilingual and code-switching ASR challenges for low resource Indian languages. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 1001–1005). volume 2.

Ko, T., Peddinti, V., Povey, D., & Khudanpur, S. (2015). Audio augmentation for speech recognition. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 3586–3589).

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 66–75). Association for Computational Linguistics volume 1.

Kumar, M., Kuriakose, J., Thyagachandran, A., Kumar, A., Seth, A., Prasad, L., Jaiswal, S., Prakash, A., & Murthy, H. (2021). Dual script E2E framework for multilingual and code-switching ASR. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 381–385). volume 1.

Lakomkin, E., Heymann, J., Sklyar, I., & Wiesler, S. (2020). Subword regularization: An analysis of scalability and generalization for end-to-end automatic speech recognition. *arXiv*, *abs/2008.04034*.

Lakshmi, A., & Murthy, H. A. (2006). A syllable based continuous speech recognizer for Tamil. In *Proc. of the 9th International Conference on Spoken Language Processing (ICSLP), Interspeech* (p. 1878 – 1881). volume 4.

Li, B., Zhang, Y., Sainath, T., Wu, Y., & Chan, W. (2019). Bytes are all you need: end-to-end multilingual speech recognition and synthesis with bytes. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5621–5625).

Panda, S. P., & Nayak, A. K. (2016). Automatic speech segmentation in syl-

lable centric speech recognition system. *International Journal of Speech Technology*, *19*, 9 – 18.

Park, D., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E., & Le, Q. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 2613–2617).

Peddinti, V., Povey, D., & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 3214–3218).

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., & Khudanpur, S. (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 2751–2755).

Prabhavalkar, R., Rao, K., Sainath, T., Li, B., Johnson, L., & Jaitly, N. (2017). A comparison of sequence-to-sequence models for speech recognition. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 939–943).

Prakash, A., Leela Thomas, A., Umesh, S., & A Murthy, H. (2019). Building multilingual end-to-end speech synthesisers for Indian languages. In *Proc. of the 10th ISCA Workshop on Speech Synthesis (SSW 10)* (pp. 194–199).

Qu, Z., Haghani, P., Weinstein, E., & Moreno, P. (2017). Syllable-based acoustic modeling with CTC-SMBR-LSTM. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 173–177).

Rao, K., Sak, H., & Prabhavalkar, R. (2017). Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer.

In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 193–199).

Saon, G., Soltau, H., Nahamoo, D., & Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 55–59).

Scharf, P. (2013). Linguistic issues and intelligent technological solutions in encoding Sanskrit. *Document Numerique*, *16*, 15–29.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 1715–1725). Association for Computational Linguistics volume 1.

Shetty, V., & Umesh, S. (2021). Exploring the use of common label set to improve speech recognition of low resource Indian languages. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7228–7232).

SLP1 mapping (2015). Indic script comparison table. URL: `https://www.sanskritlibrary.org/scriptTable.html`.

Thomas, S., Audhkhasi, K., & Kingsbury, B. (2020). Transliteration based data augmentation for training multilingual ASR acoustic models in low resource settings. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 4736–4740).

Toshniwal, S., Sainath, T. N., Weiss, R. J., Li, B., Moreno, P., Weinstein, E., & Rao, K. (2018). Multilingual speech recognition with a single end-to-end model. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4904–4908).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Watanabe, S., Hori, T., & Hershey, J. R. (2017a). Language independent end-to-end architecture for joint language identification and speech recognition. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 265–271).

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., & Ochiai, T. (2018). ESPNet: End-to-end speech processing toolkit. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 2207–2211).

Watanabe, S., Hori, T., Kim, S., Hershey, J., & Hayashi, T. (2017b). Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal on Selected Topics in Signal Processing*, *11*, 1240–1253.

Xiao, Z., Ou, Z., Chu, W., & Lin, H. (2018). Hybrid CTC-attention based end-to-end speech recognition using subword units. In *Proc. of the 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)* (pp. 146–150).

Yuan, J., Cai, X., Gao, D., Zheng, R., Huang, L., & Church, K. (2021). Decoupling recognition and transcription in Mandarin ASR. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 1019–1025).

Zeyer, A., Zhou, W., Ng, T. S. E., Schluter, R., & Ney, H. (2020). Investigations on phoneme-based end-to-end speech recognition. *arXiv*, *abs/2005.09336*.

Zhang, S., Lei, M., Liu, Y., & Li, W. (2019). Investigation of modeling units for Mandarin speech recognition using DFSMN-CTC-SMBR. In *Proc. of the*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7085–7089).

Zhao, Y., Yue, J., Xu, X., Wu, L., & Li, X. (2019). End-to-end-based Tibetan multitask speech recognition. *IEEE Access*, *7*, 162519–162529.

Zhou, S., Dong, L., Xu, S., & Xu, B. (2018). Syllable-based sequence-to-sequence speech recognition with the transformer in Mandarin Chinese. In *Proc. of the Annual Conference of the International Speech Communication Association, Interspeech* (pp. 791–795).

Zou, W., Jiang, D., Zhao, S., Yang, G., & Li, X. (2018). Comparable study of modeling units for end-to-end Mandarin speech recognition. In *Proc. of the 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)* (pp. 369–373).