INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Low Resource ASR: The surprising effectiveness of High Resource Transliteration

Shreya Khare[1†], *Ashish Mittal*[1†], *Anuj Diwan*[2†], *Sunita Sarawagi*[2], *Preethi Jyothi*[2], *Samarth Bharadwaj*[1]

[1] IBM Research, India
[2] IIT Bombay, India

skhare34@in.ibm.com

Abstract

Cross-lingual transfer of knowledge from high-resource languages to low-resource languages is an important research problem in automatic speech recognition (ASR). We propose a new strategy of transfer learning by pretraining using large amounts of speech in the high-resource language but with its text transliterated to the target low-resource language. This simple mapping of scripts explicitly encourages increased sharing between the output spaces of both languages and is surprisingly effective even when the high-resource and low-resource languages are from unrelated language families. The utility of our proposed technique is more evident in very low-resource scenarios, where better initializations are more beneficial. We evaluate our technique on a transformer ASR architecture and the state-of-the-art wav2vec2.0 ASR architecture, with English as the high-resource language and six languages as low-resource targets. With access to 1 hour of target speech, we obtain relative WER reductions of up to 8.2% compared to existing transfer-learning approaches.

Index Terms: Low resource ASR, Transliteration, Fine-tuning, Transfer learning

1. Introduction

End-to-end (E2E) systems have emerged as a de facto modeling choice for ASR in recent years and demonstrate superior performance compared to traditional cascaded ASR systems. However, E2E systems entail highly resource-intensive training with large amounts of labeled speech to perform well. This requirement tilts the balance in favour of high-resource languages like English for which large labeled speech corpora are publicly available. In contrast, for a majority of the world's languages, only limited amounts of transcribed speech are available. Improving the performance of E2E ASR systems for such low-resource languages by effectively making use of large amounts of labeled speech in high-resource languages is of great interest to the speech community.

Transfer learning techniques for speech recognition aim at effectively transferring knowledge from high-resource languages to low-resource languages and have been extensively studied [1, 2, 3, 4, 5, 6, 7, 8]. A popular paradigm for transfer learning in E2E systems is to pretrain a model on labeled speech from one (or more) high-resource languages and then fine-tune all or parts of the model on speech from the low-resource language. Often the high-resource and low-resource languages utilize very different grapheme vocabularies. Such disparities in the output vocabularies have been predominantly handled in prior work by either training only the encoder layers or training both the encoder and decoder layers in the E2E ASR system using the high-resource language. In the latter case, the output softmax layer are on the high resource graphemes and need to be replaced with a new one corresponding to the target low-resource language before fine-tuning on speech from the low-resource language. In these approaches, the sharing across languages is latent and not controllable in the output space when the language-specific graphemes are disjoint.

In this work, we propose a strategy of increased sharing in the output grapheme space by transliteration of high resource language transcription to the low-resource language. With English as the high-resource language, we adapt to six different low-resource world languages. For these languages, off-the-shelf transliteration libraries that can convert any English text to graphemes in the languages are easily available, since transliteration is a popular input typing tool for the large number of speakers of minority languages.

We use transliteration as a first step to convert transcriptions of large English speech corpora into text in the script of the target language. The E2E model is then pretrained using these transliterated transcriptions for English speech, followed by finetuning the model using limited amounts of speech in the target language. This seemingly simple technique helps the model learn a good initialization for the target language and is shown to be more effective than standard transfer learning techniques on a range of languages. Forcing English transcriptions to adopt the same script as the target language enables better sharing of model parameters, across both encoder and decoder layers. Our method is able to provide gains even with off-the-shelf, imperfect transliteration libraries since the transliterated data is used only during pre-training. In contrast the reverse approach of transliterating low resource languages to English as proposed in [9] is significantly worse since it requires a final possibly lossy transliteration back to the native language.

2. Related Work

Improving low resource ASR by exploiting labeled data from high resource languages has been an active research area starting from traditional HMM-based models [1] to modern neural systems [10, 2, 3, 4, 5, 6, 7, 8]. While some recent systems have attempted transfer using accoustic models with a shared phone layer [5] or separate phoneme layers [4] or union of the two [11], our focus here is on the more popular end-to-end systems (E2E) that predict graphemes at the last layer. Transfer learning on E2E systems using labelled data of high-resource languages have been attempted in three settings: (1) Separate softmax layers over grapheme vocabulary of each language that are trained jointly [3], (2) Pre-training on high-resource graphemes followed by fine-tuning a separate grapheme soft-

[†] contributed equally.

http://dx.doi.org/10.21437/Interspeech.2021-2062

max on the target low-resource language [2, 6], and (3) Training a shared softmax layer by taking a union of all grapheme vocabularies, usually applied when languages share graphemes [7]. In all these approaches the sharing across languages is latent and not controllable explicitly in the output space when the language-specific grapheme vocabularies are disjoint.

We attempt to remedy that by transliterating the high-resource graphemes (English) to the low-resource graphemes. While transliteration has been used extensively in improving machine translation [12], information retrieval [13] and cross-lingual tasks [14], little work has focused on improving speech recognition performance. Recently, [9] proposed the reverse transliteration from Indian languages to English and show improvement over a normal multilingual model. In this paper we show that our direction of transliteration from English provides much higher gains, and the reverse direction is often worse than earlier transfer learning approaches that do not attempt to share graphemes. Transliteration-based approaches are also relevant for code-switched ASR [15]. Concurrently with our work, [16] also proposed to pre-train multilingual models by transliterating low resource graphemes amongst one another. However, they obtain graphemes as predictions by an initial low resource ASR model when input high resource audio. The initial low resource ASR model is trained with its own limited data, and is likely to make highly noisy predictions. Pre-training with a model's own noisy predictions could introduce negative feedback. In contrast we propose a less error-prone method of harnessing the gold transcripts of high resource language via pre-existing transliteration libraries.

Another recent promising direction is learning transferable latent speech representations by pre-training a model from unlabelled speech [8]. Our transliteration of labelled data can be used to further fine-tune even such pre-trained models, and we show significant gains on a recent self-supervised wav2vec2.0 [17] pre-trained model.

3. Proposed approach

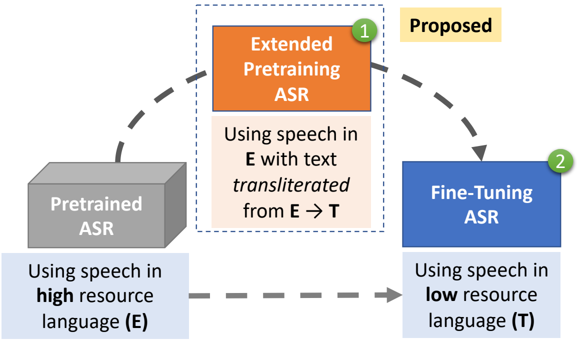
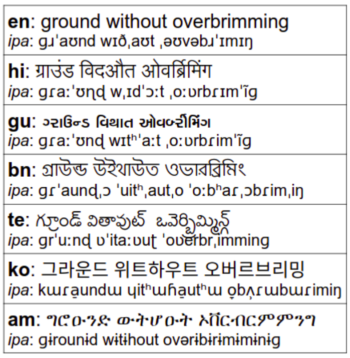

Proposed
1
Extended Pretraining ASR
Using speech in E with text *translitereated* from E → T
Pretrained ASR
Using speech in **high** resource language **(E)**
2
Fine-Tuning ASR
Using speech in **low** resource language **(T)**

Figure 1: *Training procedure for low resource languages using the proposed transliteration method*

The overall training procedure of our proposed approach is shown in Figure 1. The training consists of two stages: pretraining followed by finetuning. During pretraining, we transliterate the high-resource language text to the target low-resource language and train the ASR model using the original audio data and this transliterated text. Next, we finetune the pretrained ASR model on the target language data, after reinitializing the output layers since the output vocabulary has changed.

Transliteration is utilized to convert text from one script or language to another, often preserving sounds across languages. By supervising using transliterated text during the pretraining stage, we expect that better sound to text mappings for the target language are implicitly learned. We intend to use existing transliteration tools which supports the proposed transliterations. There are two common transliteration approaches; rule-based and machine translation. The rule-based approach relies on character mappings between the two scripts whereas machine translation approaches learn from parallel training data. To show that the proposed method works with even a simple transliteration system and is therefore easily extended to several other low resource languages, we use existing simple off the shelf systems. For the four Indian languages we used `indic-trans` [18], for Korean we used the Microsoft Azure API[1], and for Amharic, we use the Google Transliterate API via the `google-transliterate-api`[2] pip package. In fact, developing a custom phoneme-based transliteration system did not yield any improvements over using off-the-shelf systems, so we stuck with the latter. Figure 2 shows examples of English text transliterated to corresponding target languages.


en: ground without overbrimming
ipa: gɹ'aʊnd wɪð,aʊt ,əʊvəbɹ'ɪmɪŋ

hi: ग्राउंड विदऔत ओवर्ब्रिमिंग
ipa: gɾa:'ʊɳɖ w,ɪd'ɔ:t ,o:ʋrbɾɪm'ĩg

gu: ગ્રાઉન્ડ વિથાત ઓવર્બ્રીમિંગ
ipa: gɾa:'ʊnɖ wɪtʰ'a:t ,o:ʋrbɾim'ĩg

bn: গ্রাউন্ড উইথাউত ওভারব্রিমিং
ipa: gɾ'aunɖ,ɔ 'uitʰ,aut,o 'o:bʱaɾ,ɔbɾim,iŋ

te: గ్రూండ్ విథావుట్ ఒవెర్బ్రిమ్మిన్గ్
ipa: gr'u:nɖ ʋ'ita:ʋuʈ 'oʊërbr,imming

ko: 그라운드 위트하우트 오버르브리밍
ipa: kɯɾa̠undɯ ɥitʰɯɦa̠utʰɯ o̞bʌ̹ɾɯbɯɾimiŋ

am: ግሮዑንድ ውትሆዑት ኦቨርብርምምንግ
ipa: gɨrounɨd wɨtɨhout ovərɨbɨrɨmɨmɨnɨg

Figure 2: *Examples of transliteration to all 6 languages for a sample English sentence.*

4. Experiments

We evaluate our proposed approach, **Eng2Tgt**, on six languages: Hindi, Telugu, Gujarati,Bengali, Korean, Amharic and over two model architectures (ESPNet and Wav2vec 2.0) and contrast with two existing transfer-learning approaches.[3]

4.1. Baselines

We devise three natural baselines that ablate the choices made in our proposed approach: 1) **NoPre.** We train the transformer ASR models from scratch on the target language datasets i.e., randomized initialization without any pretraining. 2) **EngPre.** We first pretrain the ASR model on untransliterated English i.e. using the original Latin script. Next, we finetune on the Indian language datasets. This baseline is an established method of transfer learning as discussed in Section 2. 3) **Tgt2Eng.** This is an approach adapted from the transliterated-based multilingual modelling for multiple languages proposed in [9]. Although they address multilingual modelling, we adapt this work to an

[1] https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-transliterate

[2] https://pypi.org/project/google-transliteration-api/

[3] For Amharic and Korean, we only report wav2vec2.0 WERs; the WERs from the Transformer model were unstable possibly due to poor seeds and require further investigation.

Table 1: *Size of the various datasets in hours.*

	Train	Dev	Test
Hindi	40.2	4.97	5.02
Telugu	30.0	2.55	2.44
Gujarati	39.7	2.00	3.15
Bengali	40.0	5.00	5.00
Korean	49.7	2.00	1.19
Amharic	18.0	2.00	0.73

Table 2: *Word Error Rate (WER) across different transliteration schemes on the Transformer ASR system*

Duration	Method	Hin	Tel	Guj	Ben
Full	NoPre	16.3	29.5	19.2	36.2
	EngPre	**15.6**	26.3	17.6	27.2
	Tgt2Eng	25.2	86.4	44.2	75.5
	Eng2Tgt (Ours)	**15.6**	**25.9**	**17**	**26.2**
10 hour	NoPre	65.5	87.1	55.2	93.4
	EngPre	29.4	51.9	**33.4**	57.1
	Tgt2Eng	40.1	91.3	55.8	85.6
	Eng2Tgt (Ours)	**28**	**48.5**	34.4	**56.4**

English-centric transliteration approach to create a competitive baseline for our transliteration-based approach. We first pretrain the ASR model on English speech that uses the original Latin script. We transliterate the low-resource transcriptions to English with the help of the same transliteration modules used in our approach. Finally, we finetune on the Latin-script low-resource transcriptions. Predictions from this system will be in the Latin script. Therefore, we transliterate the hypotheses back to the low-resource language's for a fair comparison.

4.2. Datasets

For high-resource pretraining, we use the 100-hour Librispeech English dataset [19]. For the four Indian languages among the low-resource target languages, we used the Microsoft Speech Corpus (Indian Languages) dataset [20] for Gujarati and Telugu, the Hindi ASR Challenge dataset [21] for Hindi, and the OpenSLR Large Bengali dataset [22] for Bengali. We use the Zeroth Korean [23] dataset for Korean and the ALFFA Amharic [24] dataset for Amharic. Note that for Korean, the data setup was performed using the Zeroth-Korean Kaldi [25] recipe. This recipe uses a morphological segmentation tool called `morfessor` [26] to morphologically segment the provided text. All of the datasets mentioned above are monolingual, without any examples of code mixing with English. Table 1 presents detailed statistics per language. To further simulate the low resource setting we downsample the available training set to 10 hours and 1 hour.

4.3. Transformer-based E2E ASR

Experimental Setup. The Transformer ASR systems are built using the ESPNet Toolkit [27] that offers recipes to train hybrid CTC-attention end-to-end systems [28]. A Byte-Pair Encoding(BPE) [29] tokenization with a vocabulary size of 5000 is used to tokenize the transcriptions during pretraining and finetuning and 80-dimensional log-mel audio features (with pitch information) are used. The Transformer [30]-based hybrid CTC-attention model uses a CTC weight of 0.3 and an attention weight of 0.7 during both pretraining and finetuning. A 12-layer encoder network, and a 6-layer decoder network is used, each with 2048 units, with a 0.1 dropout rate. Each layer has eight 64-dimensional attention heads that are concatenated to give a 512-dimensional attention vector. During both pretraining and finetuning, models were trained for a maximum of 80 epochs with an early-stopping patience of 8 using the noam optimizer from [30] with a learning rate of 10 and 25000 warmup steps. Label smoothing and preprocessing using spectral augmentation is also used. (Note that while finetuning on the target language we transfer both encoder and decoder weights. This yielded better performance than transferring only the encoder weights, as observed in [31].) We average the best 5 models for decoding based on multiple criteria, i.e. validation loss, validation accuracy, or simply the last 5 models, with a beam size of 10 or 20 and a CTC weight of 0.3 or 0.5.

Results. Table 2 lists word error rates (WERs) on the test sets of each language for all four methods and two training durations: Full that uses the entire training set and a 10 hour subset.

From the results in Table 2 we make a number of observations. First, compared to the no pre-training baseline (NoPre), all three methods of harnessing the English high resource corpus provide significant gains. The gains are particularly striking for the 10 hour low resource setting. For every setting, the reverse transliteration approach (Tgt2Eng) is worse than existing fine-tuning (EngPre) that retains each language in its original grapheme space. In contrast, our approach is better than both these approaches in most cases, with larger observed gains in the low-resource 10-hours setting.

We offer an intuitive explanation for why one may expect Eng2Tgt to do better than EngPre. Unlike Eng2Pre, in Eng2Tgt the pretraining commits to target character labels appearing in transliterations of the English data. The fine-tuning phase then only needs to focus on learning new sounds that are missing in the English speech data. While simplistic, this explanation is somewhat borne out by the CER statistics. Of the top five characters in which Eng2Tgt gains the largest CER reductions over EngPre, most of them have very low frequencies of less than 0.5% in the transliterated corpora.

Some examples of erroneous transliterations for the Indian languages are shown in Figure 3; when a word is first transliterated into Latin script, it gets transliterated back into a different word in the target script. These errors occur because multiple Indian words can transliterate to the same English word; many orthographic differences in these Indian languages (e.g., long and short vowels, aspirated and unaspirated consonants, etc.) are lost when transliterated to English. The superior performance of our Eng2Tgt approach demonstrates that even with imperfect transliteration systems, it is possible to design effective transfer strategies. Our approach relies on the transliterated text only during the pre-training step, and the final finetuning with the target language is able to unlearn errors induced due to the imperfect transliteration. On the other hand, Tgt2Eng produces predictions in the Latin script, where the final transliteration back to the Indian language induces large errors in the output transcripts that remain uncorrected.

4.4. wav2vec2.0-based ASR

Experimental Setup. We show the promise of our proposed transliteration-based pretraining on the recently introduced self-supervised encoder architecture wav2vec2.0 [17]. We use

चीफ -> chif -> चिफ　　निर्णयों -> nirnyon -> निर्नयोन
આર્થિક -> aarthik -> આર્તિક　　પીપોદર -> pepodar -> પેપોદર

Figure 3: *Erroneous transliterations*

Table 3: *WERs on the wav2vec2.0 ASR system with 10 and 1 hour of target data. For Korean, CERs are in parentheses.*

	Method	Hin	Tel	Guj	Ben	Kor	Amh
10	SelfSup	23.8	35.7	25.2	29.4	21.79 (14.3)	**36.16**
	EngPre	24.0	37.6	25.0	32.3	**13.16** **(9.4)**	36.61
	Ours	**23.6**	**34.5**	**23.2**	**28.2**	**13.16** (9.6)	39.48
1	SelfSup	28.9	42.1	57.1	**83.1**	99.87 (83.3)	**57.99**
	EngPre	29.9	48.1	62.1	92.3	66.36 (40.8)	60.7
	Ours	**28.5**	**41.5**	**55.2**	88.9	**62.08** **(37.2)**	59.39

the pretrained wav2vec2.0 model estimated using unsupervised pretraining on the complete Librispeech dataset and refer to this system as SelfSup. With SelfSup as our starting point, we report numbers for both EngPre (involving supervised pretraining with 100 hours of English speech) and Eng2Tgt that applies transliteration to the 100 hours of English data before pretraining. (Tgt2Eng was excluded due to its poor performance on the Transformer ASR experiments.) For finetuning on our datasets, we use the same training recipe as described in [17] for the 10 hr and 1 hr settings. We perform inference using the default hyperparameters as described in [17] without an LM for all languages except Korean. For Korean, an LM was necessary during decoding to obtain meaningful WERs. We use a 4-gram KenLM [32] estimated on the training set. Since we start with a powerful pretrained model, we show results in Table 3 for very low-resource scenarios i.e. using 10-hour and 1-hour training subsets.

Results. SelfSup offers a much stronger baseline compared to NoPre in Table 2. We observe that our proposed pre-training with transliterated data provides gains even on a system like wav2vec that leverages powerful pretrained models. Our method also outperforms EngPre in most settings. For most languages in both settings, the EngPre baseline is worse than even the SelfSup baseline, despite being pretrained on more (Latin-script English) data. A major exception is Amharic. We will investigate reasons for this difference in Section 4.5.

4.5. Analysis and discussions

Here we attempt to understand the conditions under which our proposed approach is likely to help. Two properties need to *simultaneously* hold for pretraining with transliterated English to be useful: (1) The transliteration library needs to be acoustically consistent, and (2) The overlap in the sounds (phones) of the high and low-resource language has to be high, that is the phonological distance between the two languages should be low. We analyze how well these two properties hold.

Acoustic consistency of transliterations We measure acoustic consistency of the various transliteration libraries using the following PER (phone error rate)-based method. First we convert the Latin-script Librispeech English training set text to IPA using an English g2p tool, `epitran` [33] and also convert the transliterated English to IPA using native-language g2p tools. We use `epitran` [33] for Hindi, Telugu, Bengali and Amharic and UIUC's `g2ps`[4]. We could not find a reliable Korean g2p tool. We then compute the PER between these two IPA sequences ignoring space tokens. Table 4 shows the PER of various languages. We observe that Amharic has a relatively high error compared to languages such as Hindi and Gujrati.

Table 4: *Second row shows acoustic consistency of transliteration measured as the phone error rate between transliterated and English text. Third row shows the KL divergence between the unigram phone distribution of English and native language.*

Language	am	bn	hi	te	gu
Transliteration PER	89	90	76	82	72
KL dist phones	8.2	13.6	10.2	11.4	15.6

Table 5: *WERs for Gujarati comparing Eng2Tgt pretraining using 40 hrs of transliterated Hindi vs. transliterated English.*

	Full	10 Hours
Hin2Tgt	**18.3**	**35.4**
Eng2Tgt40	21.3	38.1

Phonological similarity of low resource language with English Many different methods have been proposed for measuring the phonological similarity of two languages [34]. Here we resort to a comparison of a simple unigram distribution of the phones in English and native language. The third row of Table 4 shows the results. We observe that for languages like Hindi and Telugu where the KL distance in phone distribution is small *and* the transliteration PER is low, we get consistent gains across different architectures and training data sizes. This analysis provides a partial explanation of our observed numbers but for languages like Amharic more investigation is required.

Effect of related languages Table 5 shows WERs for Gujarati in the full and 10 hour training settings on the Transformer system. Using our approach, we compare performance when using a language related to the target language (e.g. Hindi) (Hin2Tgt) during pretraining as opposed to English. The transliterated English data during pretraining was reduced to 40 hours to match the amount of transliterated Hindi pretraining data (Eng2Tgt40). We observe there's a significant improvement in performance across both training settings with using Hindi instead of English during pretraining. Using both transliterated English and Hindi data during pretraining for the 10-hour Gujarati task further reduces WERs from 55.8% to 32.4%. This opens up interesting directions to explore as future work.

5. Conclusion

This work explores the surprisingly effective role of transliteration in training an E2E ASR system. We propose a simple transliteration-based transfer learning technique easily adaptable to other low-resource languages and demonstrate its utility on two state-of-the-art ASR systems, showing significant improvements in performance over established transfer-learning approaches despite using an imperfect transliteration system.

[4] https://github.com/uiuc-sst/g2ps

6. References

[1] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, no. 1-2, pp. 31–51, 2001.

[2] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP 2013*. IEEE, 2013, pp. 7319–7323.

[3] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013, pp. 7304–7308.

[4] S. Tong, P. N. Garner, and H. Bourlard, "Multilingual training and cross-lingual adaptation on CTC-based acoustic model," *arXiv:1711.10025*, 2017.

[5] X. Li, S. Dalmia, J. Li, M. Lee, P. Littell, J. Yao, A. Anastasopoulos, D. R. Mortensen, G. Neubig, A. W. Black *et al.*, "Universal phone recognition with a multilingual allophone system," in *ICASSP*, 2020, pp. 8249–8253.

[6] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, "Transfer learning for speech recognition on a budget," *arXiv preprint arXiv:1706.00290*, 2017.

[7] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *ICASSP 2018*. IEEE, 2018, pp. 4904–4908.

[8] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-lingual Representation Learning for Speech Recognition," 2020.

[9] A. Datta, B. Ramabhadran, J. Emond, A. Kannan, and B. Roark, "Language-Agnostic Multilingual Modeling," in *ICASSP*, 2020, pp. 8239–8243.

[10] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *APSIPA 2015*. IEEE, 2015, pp. 1225–1237.

[11] O. Scharenborg, F. Ciannella, S. Palaskar, A. Black, F. Metze, L. Ondel, and M. Hasegawa-Johnson, "Building an asr system for a low-resource language through the adaptation of a high-resource language asr system: Preliminary results," *Proceedings of ICNLSSPo*, 2017.

[12] V. Goyal, S. Kumar, and D. M. Sharma, "Efficient neural machine translation for low-resource languages via exploiting related languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Online: Association for Computational Linguistics, Jul. 2020.

[13] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, and M. Shrivastava, "Iiit-h system submission for fire2014 shared task on transliterated search," in *Proceedings of the Forum for Information Retrieval Evaluation*, ser. FIRE '14, 2015.

[14] Y. Khemchandani, S. Mehtani, V. Patil, A. Awasthi, P. Talukdar, and S. Sarawagi, "Exploiting language relatedness for low web-resource language model adaptation: An indic languages study," in *ACL-IJCNLP*, 2021.

[15] J. Emond, B. Ramabhadran, B. Roark, P. Moreno, and M. Ma, "Transliteration Based Approaches to Improve Code-Switched Speech Recognition Performance," in *SLT 2018*, 2018, pp. 448–455.

[16] S. Thomas, K. Audhkhasi, and B. Kingsbury, "Transliteration based data augmentation for training multilingual ASR acoustic models in low resource settings," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 4736–4740.

[17] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[18] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, and M. Shrivastava, "IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search," in *FIRE*, 2015, pp. 48–53. [Online]. Available: http://doi.acm.org/10.1145/2824864.2824872

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP 2015*. IEEE, 2015, pp. 5206–5210.

[20] B. Srivastava, S. Sitaram, R. Mehta, K. Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, "Interspeech 2018 low resource automatic speech recognition challenge for indian languages," 08 2018, pp. 11–14.

[21] S. L. I. Madras, "Hindi ASR Challenge Dataset," July 2020. [Online]. Available: https://github.com/Speech-Lab-IITM/Hindi-ASR-Challenge

[22] O. Kjartansson, S. Sarin, K. Pipatsrisawat, M. Jansche, and L. Ha, "Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali," in *Proc. of 6th SLTU*, 2018, pp. 52–55.

[23] "Zeroth-korean," https://www.openslr.org/40/.

[24] S. T. Abate, W. Menzel, and B. Tafila, "An amharic speech corpus for large vocabulary continuous speech recognition," in *INTERSPEECH-2005*, 2005.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[26] S. Virpioja, P. Smit, S.-A. Grönroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," D4 Julkaistu kehittämis- tai tutkimusraportti tai -selvitys, 2013. [Online]. Available: http://urn.fi/URN:ISBN:978-952-60-5501-5

[27] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E Y Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[28] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *Proc. of ACL*, 2016. [Online]. Available: http://dx.doi.org/10.18653/v1/P16-1162

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *NeurIPS*, 2017.

[31] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiat, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *SLT 2018*. IEEE, 2018, pp. 521–527.

[32] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, Jul. 2011, pp. 187–197. [Online]. Available: https://www.aclweb.org/anthology/W11-2123

[33] D. R. Mortensen, S. Dalmia, and P. Littell, "Epitran: Precision G2P for many languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Paris, France: European Language Resources Association (ELRA), May 2018.

[34] S. E. Eden, "Measuring phonological distance between languages" (ucl (university college london)," Ph.D. dissertation, 2018.