MUCS 2021: Multilingual and code-switching ASR challenges for low resource Indian languages

Anuj Diwan[1], Rakesh Vaideeswaran[2], Sanket Shah[3], Ankita Singh[1], Srinivasa Raghavan[4], Shreya Khare[5], Vinit Unni[1], Saurabh Vyas[4], Akash Rajpuria[4], Chiranjeevi Yarra[6], Ashish Mittal[5], Prasanta Kumar Ghosh[2], Preethi Jyothi[1], Kalika Bali[3], Vivek Seshadri[3], Sunayana Sitaram[3], Samarth Bharadwaj[5], Jai Nanavati[4], Raoul Nanavati[4], Karthik Sankaranarayanan[5],

[1] Computer Science and Engineering, Indian Institute of Technology (IIT), Bombay, India
[2] Electrical Engineering, Indian Institute of Science (IISc), Bangalore 560012, India
[3] Microsoft Research India, Hyderabad, India
[4] Navana Tech India Private Limited, Bangalore, India
[5] IBM Research India, Bangalore, India
[6] Language Technologies Research Center (LTRC), IIIT Hyderabad, 500032, India

`is21_ss_indicasr@iisc.ac.in`

Abstract

Recently, there is an increasing interest in multilingual automatic speech recognition (ASR) where a speech recognition system caters to multiple low resource languages by taking advantage of low amounts of labelled corpora in multiple languages. With multilingualism becoming common in today's world, there has been increasing interest in code-switching ASR as well. In code-switching, multiple languages are freely interchanged within a single sentence or between sentences. The success of low-resource multilingual and code-switching (MUCS) ASR often depends on the variety of languages in terms of their acoustics, linguistic characteristics as well as the amount of data available and how these are carefully considered in building the ASR system. In this MUCS 2021 challenge, we would like to focus on building MUCS ASR systems through two different subtasks related to a total of seven Indian languages, namely Hindi, Marathi, Odia, Tamil, Telugu, Gujarati and Bengali. For this purpose, we provide a total of $\sim$600 hours of transcribed speech data, comprising train and test sets, in these languages, including two code-switched language pairs, Hindi-English and Bengali-English. We also provide baseline recipes[1] for both the subtasks with 30.73% and 32.45% word error rate on the MUCS test sets, respectively.

Index Terms: Multilingual, Code-switching, low-resource

1. Introduction

India is a country of language continuum, where every few kilometres, the dialect/language changes [1]. Various language families or genealogical types have been reported, in which the vast number of Indian languages can be classified, including Austro-Asiatic, Dravidian, Indo-Aryan, Tibeto-Burman and more recently, Tai-Kadai and Great Andamanese [2, 3]. However, there are no boundaries among these language families; rather, languages across different language families share linguistic traits, including retroflex sounds, absence of prepositions and many more resulting in acoustic and linguistic richness. According to the 2001 census, 29 Indian languages have more than a million speakers. Among these, 22 languages have been given the official language status by the Government of India [4, 5]. Most of these languages are low resource and do not have a written script. Hence, speech technology solutions, such as automatic speech recognition (ASR), would greatly benefit such communities [6]. Another common linguistic phenomenon in multilingual societies is code-switching [7], typically between an Indian language and (Indian) English. Understanding code-switching patterns in different languages and developing accurate code-switching ASR remain a challenge due to the lack of large code-switched corpora [8, 9].

In such resource-constrained settings, exploiting unique properties and similarities among the Indian languages could help build multilingual and code-switching (MUCS) ASR systems. Prior works have shown that multilingual ASR systems that leverage data from many languages could explore common acoustic properties across similar phonemes or graphemes [10, 11, 12, 13, 14]. This is achieved by gathering a large amount of data from multiple low-resource languages. Also, multilingual ASR strategies are effective in exploiting the code-switching phenomena in the speech of the source languages [15]. However, there is an emphasis on the need for the languages' right choice for better performance [16], as significant variations between the languages could degrade the ASR performance under multilingual scenarios [12]. In such cases, a dedicated monolingual ASR could perform better even with lesser speech data than a multilingual [17, 18, 19] or code-switching ASR.

Considering the factors above, in this MUCS 2021 challenge, we have selected six Indian languages, Hindi, Marathi, Odia, Telugu, Tamil and Gujarati, for multilingual ASR; and two code-switched language pairs, Hindi-English and Bengali-English, for code-switching ASR. Unlike prior works on multilingual ASR, the languages selected 1) consider the influences of three major language families – Indo-Aryan, Dravidian and Austro-Asiatic, which influences most of the Indian languages [4], 2) cover four demographic regions of India – East, West, South and North, and, 3) ensure continuum across languages. It is expected that a multilingual ASR built on these languages could be helpful to extend to other low-resource languages [6]. Further, most of the multilingual ASR works have considered languages other than Indian languages. Works that consider the Indian languages, however, use data that is either not publicly available or limited in size [5, 20, 6, 17, 21]. This is similarly true for code-switched speech, and prior work has predominantly focused on Hindi-English [22, 23, 24, 25, 26]. MUCS 2021 challenge significantly contributes in this context, as we provide a larger corpus ($\sim$600 hours of transcribed speech from

Additional authors: Tejaswi Seeram and Basil Abraham, Microsoft Corporation India.

[1] `https://github.com/navana-tech/baseline_recipe_is21s_indic_asr_challenge`

Table 1: *Details of multilingual ASR train (Trn), test (Tst) and blind test (Blnd) data – size, channel compression (Ch.comp), number of unique sentences (Uniq sent), number of speakers (Spkrs) and vocabulary size in words (vocab). All six languages' audio files are single-channel and encoded in 16-bit with a sampling rate of 8kHz except for train and test set of Telugu, Tamil and Gujarati, at 16kHz.*

	Hindi			Marathi			Odia			Telugu			Tamil			Gujarati		
	Trn	Tst	Blnd	Trn	Tst	Blnd	Trn	Tst	Blnd	Trn	Tst	Blnd	Trn	Tst	Blnd	Trn	Tst	Blnd
Size (hrs)	95.05	5.55	5.49	93.89	5	0.67	94.54	5.49	4.66	40	5	4.39	40	5	4.41	40	5	5.26
Ch.comp	3GP	3GP	3GP	3GP	3GP	M4A	M4A	M4A	M4A	PCM	PCM	PCM	PCM	PCM	PCM	PCM	PCM	PCM
Uniq sent	4506	386	316	2543	200	120	820	65	124	34176	2997	2506	30329	3060	2584	20257	3069	3419
Spkrs	59	19	18	31	31	-	-	-	-	464	129	129	448	118	118	94	15	18
Vocab (words)	6092	1681	1359	3245	547	350	1584	334	334	43270	10859	9602	50124	12279	10732	39428	10482	11424

different domains) compared to the existing publicly available data for Indian languages.

MUCS 2021 challenge comprises two subtasks. *Subtask1* involves building a multilingual ASR system in six languages: Hindi, Marathi, Odia, Telugu, Tamil, and Gujarati. The blind test set comprises recordings from all the six languages. *Subtask2* involves building a code-switching ASR system separately for Hindi-English and Bengali-English code-switched pairs. The blind test set comprises recordings from these two code-switched language pairs. Baseline systems are developed considering hybrid DNN-HMM models for both the subtasks and an end-to-end model for *Subtask2*. Baseline word error rates (WERs) averaged over languages on the test set and blind test set are found to be 30.73% & 32.73%, respectively, for *subtask1*. Similarly, WERs, averaged between two code-switching language pairs, for *Subtask2* are 33.35% & 28.52, 29.37% & 32.09% and 28.45% & 34.08% on test & blind sets with GMM-HMM, TDNN and end-to-end systems, respectively.

2. Details of the two Subtasks

2.1. Subtask1: Multilingual ASR

Subtask1 is for developing robust multilingual systems in six Indian languages using $\sim$450 hours of data released as a part of MUCS 2021 challenge under diversified conditions.

2.1.1. Dataset Description

Table 1 shows the data details for the Multilingual ASR specific to each language. The number of unique sentences are least in Odia data and are collected in the domains of agriculture, healthcare and finance from four districts as a representative of four different dialect regions – Sambalpur (North-Western Odia), Mayurbhanj (North Eastern Odia), Puri(Central and Standard Odia) and Koraput (Southern Odia). The Telugu, Tamil and Gujarati data are taken from Interspeech 2018 low resource ASR challenge for Indian languages, for which, the data was provided by SpeechOcean.com and Microsoft [27]. Further, in all the six languages, the percentage of out-of-vocabulary (OOV) between train & test and train & blind test are found to be in the range 17.2% to 32.8% and 8.4% to 31.1%, respectively. Also, the grapheme set in the data follows the Indian language speech sound label set (ILSL12) standard [28]. The total number of graphemes are 69, 61, 68, 64, 50 and 65 respectively for Hindi, Marathi, Odia, Telugu, Tamil and Gujarati, out of which a total number of diacritic marks in the respective languages are 16, 16, 16, 17, 7 and 17.

2.1.2. Characteristics of the dataset

The Hindi, Marathi and Odia data are collected from the respective native speakers using a reading task. For the data collection, the speakers and the text are selected to cover different language variations for better generalizability. The speakers of Hindi and Marathi belong to a high-literacy group. On the other hand, the speakers of Odia belong to a semi-literate group. The text data of Hindi and Marathi is collected from storybooks. In addition to the speaker and text variability, nativity, pronunciations and accent variations are also present in the datasets. The Odia data went through a manual check, while Hindi and Marathi speech data were passed through a semi-automatic validation process using the ASR pipeline (For more details, please refer to Section 1 of the supplementary material or Section 2.1.2 of the extended version of this paper). The train and test sets of Telugu, Tamil and Gujarati are considered as-is for MUCS 2021 challenge; however, for the construction of the blind test set, the measurement set is used and part of it is modified with speed perturbations randomly between 1.1 to 1.4 (with increments of 0.05), and/or adding one noise randomly from white, babble, and three noises chosen from the Musan dataset [29] at various the signal-to-noise ratio randomly selected from a set between 18dB to 30dB at the step of 1dB. This modification is done randomly on 29.0%, 23.8% and 34.1% of Telugu, Tamil and Gujarati measurement data, respectively.

Table 2: *Details of code-switching ASR train (Trn), test (Tst) and blind test (Blnd) data – size, uniq sent, spkrs and vocab.*

	Hin-Eng			Ben-Eng		
	Trn	Tst	Blnd	Trn	Tst	Blnd
Size (hrs)	89.86	5.18	6.24	46.11	7.02	5.53
Uniq sent	44249	2890	3831	22386	3968	2936
Spkrs	520	30	35	267	40	32
Vocab (words)	17830	3212	3527	13645	4500	3742

2.1.3. Evaluation criteria

From the channel compression schemes in Table 1, it is observed that there is a mismatch in the channel compression between train/test and blind test for Marathi. Thus, for the evaluation on the blind test, we consider both the channel matched and mismatched scenarios, for which WER across languages within each scenario is calculated. Thus, we get two WERs: 1) averaged WER across all six languages (channel mismatched scenario), 2) averaged WER across all six languages except Marathi (channel matched scenario).

2.2. Subtask2: Code-switching ASR

Subtask2 is on developing code-switching ASR on the Hindi-English and Bengali-English language pairs taken from spoken tutorials.

2.2.1. Dataset Description

The tutorials in the *Subtask2* data cover a range of technical topics, and the code-switching predominantly arises from the technical content of the lectures. The segments file in the baseline recipe provides sentence time-stamps. These time-stamps were used to derive segments from the audio file to be aligned with the transcripts given in the text file. Table 2 shows the details of the data considered for *Subtask2*. All the audio files in both datasets are sampled at 16 kHz, 16 bits encoding. The test-train sentence overlap in Hindi-English and Bengali-English data are 33.9% and 10.8%, whereas the blind test-train sentence overlaps are 2.1% and 2.9%, respectively. Speaker information for both these datasets was not available. However, we do have information about the underlying tutorials from which each sentence is derived. We assumed that each tutorial comes from a different speaker; these are the numbers reported in Table 2. The percentage of OOV words encountered in the test and blind-test for Hindi-English is 12.5% & 19.6% and for Bengali-English is 22.9% & 27.3% respectively.

2.2.2. *Characteristics and Artefacts in the Dataset*

The transcriptions in the *Subtask2* data include mathematical symbols and other technical content. It is to be noted here that these tutorials were not explicitly created for ASR but end-user consumption as videos of tutorials in various Indian languages; specifically, in our case, the transcriptions were scripts for video narrators. Thus, there are the following sources of noise in the transcriptions – 1) misalignments between transcription and its respective segment start and end times, 2) inconsistencies in the transcriptions' language for the same audio segment, 3) punctuation's enunciation in the speech, 4) language mixing within a word, 5) incomplete audio at the begin or the end of an utterance, and 6) merged English words without word boundary markings (For more details, please refer to Section 2 of the supplementary material or Section 2.2.2 of the extended version).

2.2.3. *Evaluation criteria*

To handle the transcriptions' language's inconsistencies during the evaluation, we consider transliterated WER (T-WER) besides the standard WER. To ensure that remaining noises are eliminated, we perform manual validation on the blind test set data. While the standard WER only counts an ASR hypothesis as correct if it is an exact match with the word in the reference text, T-WER counts an English word in the reference text as correctly predicted if it is in English transliterated form in the native script. To compute T-WER, we manually annotate the blind test reference text such that every English word only appeared in the Latin script. Following this, we transliterate every English word in the reference transcriptions using Google's transliteration API and manually edit them to remove valid Hindi words and fix any transliteration errors. This yielded a list of English to native script mappings and used this mapping file in the final T-WER to map English words to their transliterated forms.

3. Details of baseline schemes

3.1. Experimental setup

3.1.1. *Multilingual ASR*

Hybrid DNN-HMM: ASR model is built using the Kaldi toolkit with a sequence-trained time-delay neural network (TDNN) architecture using the lattice-free MMI objective function [30]. We consider an architecture comprising 6 TDNN blocks with a dimensionality of size 512.
Lexicon: A single lexicon is used containing the combined vocabulary of all six languages. For each language, the lexicon's entries are obtained automatically, considering a rule-based system that maps graphemes to phonemes. For the mapping, we consider the Indian speech sound label set (ILSL2) [28].
Language model (LM): A single LM is built considering the text transcriptions belonging to the train set from all six languages. For the LM, we consider a 3-gram language model developed in Kaldi using the IRSTLM toolkit. Since the LM has paths that contain multiple languages, the decoded output could result in code-mixing across the six languages.

In addition to the multilingual ASR, we also provide monolingual ASR systems' performance considering language-specific training data, lexicon and LM built with language-specific train text transcriptions.

3.1.2. *Code-switching ASR*

Hybrid DNN-HMM: The ASR model is built using the Kaldi toolkit with the same model architecture for both Hindi-English and Bengali-English language pairs. We use MFCC acoustic features to build speaker-adapted GMM-HMM models. Similar to *Subtask1*, we also build hybrid DNN-HMM ASR systems using TDNNs comprising 8 TDNN blocks with dimension 768.
End-to-end ASR: The hybrid CTC-attention model based on Transformer [31] is used with a CTC weight of 0.3 and an attention weight of 0.7. 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 contains eight 64-dimensional attention heads, which are concatenated to form a 512-dimensional attention vector. Models are trained for a maximum of 40 epochs with early-stopping patience of 3 using the Noam optimizer from [31] with a learning rate of 10 and 25000 warmup steps. Label smoothing and preprocessing using spectral augmentation is also used. The top 5 models with the best validation accuracy are averaged, and this averaged checkpoint is used for decoding. Decoding is performed with a beam size of 10 and a CTC weight of 0.4.
Lexicon: Two different lexicons are used, each for Hindi-English and Bengali-English language pair. For each lexicon, the pronunciations are generated as follows for the entire vocabulary in the respective training set. If the word is in the Devanagari/Bengali script, we consider the respective pronunciation as the word's character sequence. This is because both languages have phonetic orthographies. To obtain pronunciations for English words, we use an open-source g2p package[2]. This package provides pronunciations for numericals, retrieves pronunciations from CMUDict dictionary [32] for words that appear in its vocabulary and predict new pronunciations for words that do not. We also obtain pronunciations for the punctuations by mapping to their corresponding English words.
Language model: Two separate language models are built for each language pair. We consider a trigram language model with Kneser-Ney discounting for each LM training using the SRILM toolkit developed in Kaldi [33].

3.2. Baseline results

3.2.1. *Multilingual ASR*

Table 3 shows the WERs obtained on test and blind test sets for each of the six languages along with averaged WER across all six languages. The table shows that the WER obtained with multilingual ASR is lower for Tamil. Though the WER from the multilingual ASR system is higher in the remaining languages compared to their monolingual counterpart, it does not require any explicit language identification (LID) system. Further, it is known that multilingual ASR is effective in obtaining a better acoustic model (AM) by exploring common properties among the multiple languages. However, the multilingual ASR performance also depends on the quality of the language model, which, in this work, could introduce noise due to code-mixing of words. In order to know these variabilities, we analyse the multilingual ASR considering the code-mix in the decoded output and the AM likelihoods separately.
Analysis: Table 4 shows the amount of code-mix across the languages by averaging the percentage of words per sentence of a

[2]https://github.com/Kyubyong/g2p

Table 3: *Performance (WER in %) of multilingual and monolingual ASRs on test (Tst) and blind (Blnd) test. Averaged WER across five languages on the blind test for Multi and Mono are 33.47% and 29.98% respectively.*

		Hindi	Marathi	Odia	Tamil	Telugu	Gujarati	Avg
Multi	Tst	40.41	22.44	39.06	33.35	30.62	19.27	30.73
	Blnd	37.20	29.04	38.46	34.09	31.44	26.15	32.73
Mono	Tst	31.39	18.61	35.36	34.78	28.71	18.23	27.85
	Blnd	27.45	20.41	31.28	35.82	29.35	25.98	28.38

Table 4: *Averaged languages' (in column) code-mix word percentage in the decoded output from the multilingual ASR for the utterances belonging to a language (in row).*

	Hindi	Marathi	Odia	Tamil	Telugu	Gujarati
Hindi	82.3	0.4	0.2	0.8	1.9	14.4
Marathi	16.7	71.8	2.8	0.6	2.0	8.8
Odia	0.4	0.1	96.1	0.7	1.3	1.5
Tamil	0.1	0.0	0.0	98.4	1.0	0.4
Telugu	0.2	0.1	0.0	0.7	97.9	1.2
Gujarati	0.3	0.1	0.0	1.0	0.8	97.7

Table 5: *Averaged (std) AM log-likelihoods on test sets from monolingual and multilingual ASRs using forced-alignment.*

	Hindi	Marathi	Odia	Tamil	Telugu	Gujarati
Multi	2.5 (0.3)	2.5 (0.3)	2.6 (0.3)	2.3 (0.2)	2.2 (0.2)	2.0 (0.2)
Mono	1.9 (0.2)	1.7 (0.2)	2.4 (0.3)	2.2 (0.2)	2.1 (0.2)	1.8 (0.1)

language in the column in the decoded output of the utterances belonging to the language in the row. The higher values in diagonal entries in the table indicate the multilingual ASR's effectiveness in decoding the target language's utterance. However, the off-diagonal values of averaged percentage of words are also significant, which could be cause for higher WER with the multilingual ASR system compared to the monolingual ASR systems. Further, to know the effectiveness of AM only, we compute average (standard deviation (std)) of the AM likelihoods considering the forced-alignment process with multi and monolingual ASR models across all utterances. These are shown in Table 5. The higher likelihoods with multilingual ASR indicate its benefit over monolingual AM. Thus, the multilingual ASR performance could improve with an effective LM.

3.2.2. Code-switching ASR

Table 6 shows WERs for both the Hindi-English and Bengali-English datasets. As mentioned in Section 2.2.2, there are misalignments between the transcriptions and the timestamps in some of the training files. We present results using the original alignments that we obtained with the transcriptions (labelled as UnA). In an attempt to fix the misalignment issues, we also force-align the training files at the level of the entire tutorial with its complete transcription and recompute the segment timestamps. We retrain our systems using these re-aligned training files (labelled as ReA). As expected, we observe that the averaged ReA WERs are consistently better than the UnA WERs. (Improvements with ReA are much larger for Tst than Blnd, since the latter was manually corrected for alignment errors unlike Tst.) While the Kaldi TDNN-based system gives better WERs for the test set, the speaker adapted triphone GMM-HMM model performs the best on the blind test set.

Table 7 shows the corresponding WERs and T-WERs for the realigned (ReA) blind test sets. T-WER, being a more relaxed evaluation metric, is always better than WER. The Hindi-English code mixed data yields improved WERs evidently due to fewer OOVs and larger amounts of training data. The corresponding values for the blind-set also improve further as we calculate the transliterated scores as discussed in 2.2.2.

Table 6: *WERs from GMM-HMM, DNN-HMM and end-to-end ASR systems for Hin-Eng and Ben-Eng test (Tst) and blind-test (Blnd) sets. (ReA) and (UnA) refers to re-aligned and unaligned audio files, respectively.*

	Kaldi-Based				End-to-End	
	GMM-HMM		TDNN		Transformer	
	Tst	Blnd	Tst	Blnd	Tst	Blnd
Hin-Eng (UnA)	44.30	25.53	36.94	28.90	27.7	33.65
Ben-Eng (UnA)	39.19	32.81	34.31	35.52	37.2	43.94
Avg (UnA)	41.75	29.17	35.63	32.21	32.45	38.80
Hin-Eng (ReA)	31.56	24.66	28.40	29.03	25.9	31.19
Ben-Eng (ReA)	35.14	32.39	30.34	35.15	31.0	36.97
Avg (ReA)	33.35	28.52	29.37	32.09	28.45	34.08

Table 7: *WER and T-WER values for Kaldi and end-to-end based architectures obtained after using aligned (ReA) segments for both Hin-Eng and Ben-Eng.*

	Kaldi-Based				End-to-End	
	GMM-HMM		TDNN		Transformer	
	WER	T-WER	WER	T-WER	WER	T-WER
Hin-Eng	24.66	22.72	29.03	26.20	31.19	29.80
Ben-Eng	32.39	31.42	35.15	33.39	36.97	36.00
Avg	28.52	27.07	32.09	29.79	34.08	32.9

Table 8: *Relative substitution/deletion error rates (Err) and transliterated error rates (T-Err) of English words for Kaldi and end-to-end based architectures for Hin-Eng and Ben-Eng.*

	Kaldi-Based				End-to-End	
	GMM-HMM		TDNN		Transformer	
	Err	T-Err	Err	T-Err	Err	T-Err
Hin-Eng	62.23	59.13	60.41	56.43	57.96	56.32
Ben-Eng	58.44	56.72	55.16	52.77	56.29	54.97

Analysis: Table 8 shows the relative errors of English words in the reference transcripts either being substituted or deleted with respect to the total errors. The Hindi-English and Bengali-English blind test transcriptions contain 34.2% and 33.6% English words, respectively. The relative errors in Table 8 (all greater than 50%) show that the errors on English words are relatively more compared to Bengali/Hindi words. While the Kaldi-based GMM-HMM (tri4b) models give the best WERs on the blind test sets in Table 7, it has the highest relative error rates compared to the end-to-end and TDNN architectures as shown in Table 8.

4. Conclusion

This paper presents the dataset details and baseline recipe and results for MUltilingual and Code-Switching ASR challenges for low resource Indian languages, 2021 (MUCS 2021). MUCS 2021 challenge involves two subtasks dealing with 1) multilingual ASR and 2) code-switching ASR. Through MUCS 2021 challenge, the participants have the opportunity to address two critical challenges specific to multilingual societies, particularly in the Indian context – data scarcity and the code-switching phenomena. Through MUCS 2021 challenge, we also provide a total of $\sim$600 hours of transcribed speech data, which is a reasonably large corpus for six different Indian languages (especially when compared to the existing publicly available datasets for Indian languages). Baseline ASR systems have been developed using hybrid DNN-HMM and end-to-end models. Furthermore, carefully curated held-out blind test sets are also released to evaluate the participating teams' performance.

5. References

[1] *Ministry of Human Resource Development, India:*, Read on to know more about Indian languages, Online, Last accessed on 03-04-21.

[2] J. Heitzman and R. L. Worden, *India: A country study*. Federal Research Division, 1995.

[3] *Office of the Registrar General & Census Commissioner India, Part A:*, Family-wise grouping of the 122 Scheduled and Non-Scheduled Languages–2001, Online, Last accessed on 03-04-21.

[4] *Office of the Registrar General & Census Commissioner India, Part A:*, Distribution of the 22 scheduled languages - India, States & Union Territories - 2001 Census, , Online, Last accessed on 03-04-21.

[5] H. B. Sailor and T. Hain, "Multilingual speech recognition using language-specific phoneme recognition as auxiliary task for Indian languages," *in Interspeech*, pp. 4756–4760, 2020.

[6] A. Datta, B. Ramabhadran, J. Emond, A. Kannan, and B. Roark, "Language-agnostic multilingual modeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8239–8243.

[7] P. Auer, *Code-switching in conversation: Language, interaction and identity*. Routledge, 2013.

[8] Ö. Çetinoğlu, S. Schulz, and N. T. Vu, "Challenges of computational processing of code-switching," *arXiv preprint arXiv:1610.02213*, 2016.

[9] S. Sitaram, K. R. Chandu, S. K. Rallabandi, and A. W. Black, "A survey of code-switched speech and language processing," *arXiv preprint arXiv:1904.00784*, 2019.

[10] Y.-C. Chen, J.-Y. Hsu, C.-K. Lee, and H.-y. Lee, "DARTS-ASR: Differentiable architecture search for multilingual speech recognition and adaptation," *arXiv preprint arXiv:2005.07029*, 2020.

[11] S. Tong, P. N. Garner, and H. Bourlard, "An investigation of multilingual ASR using end-to-end LF-MMI," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6061–6065.

[12] 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 *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018, pp. 4904–4908.

[13] H. Krishna, K. Gurugubelli, A. Vuppala *et al.*, "An exploration towards joint acoustic modeling for Indian languages: IIIT-H submission for low resource speech recognition challenge for Indian languages, INTERSPEECH 2018," in *Interspeech*, 2018, pp. 3192–3196.

[14] H. Lin, L. Deng, D. Yu, Y.-f. Gong, A. Acero, and C.-H. Lee, "A study on multilingual acoustic modeling for large vocabulary ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 4333–4336.

[15] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5621–5625.

[16] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, E. Kislal, L. Mangu, M. Nussbaum-Thom, M. Picheny *et al.*, "Multilingual representations for low resource speech recognition and keyword search," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 259–266.

[17] V. Pratap, A. Sriram, P. Tomasello, A. Hannun, V. Liptchinsky, G. Synnaeve, and R. Collobert, "Massively multilingual ASR: 50 languages, 1 model, 1 billion parameters," *arXiv preprint arXiv:2007.03001*, 2020.

[18] M. Miiller, S. Stiiker, and A. Waibel, "Multilingual adaptation of RNN based ASR systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5219–5223.

[19] R. Gretter, "Euronews: a multilingual benchmark for ASR and LID," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[20] K. Manjunath, K. S. Rao, D. B. Jayagopi, and V. Ramasubramanian, "Indian languages ASR: A multilingual phone recognition framework with IPA based common phone-set, predicted articulatory features and feature fusion." in *Interspeech*, 2018, pp. 1016–1020.

[21] C. Liu, Q. Zhang, X. Zhang, K. Singh, Y. Saraf, and G. Zweig, "Multilingual graphemic hybrid ASR with massive data augmentation," *arXiv preprint arXiv:1909.06522*, 2019.

[22] A. Dey and P. Fung, "A Hindi-English code-switching corpus." in *LREC*, 2014, pp. 2410–2413.

[23] A. Pandey, B. M. L. Srivastava, and S. V. Gangashetty, "Adapting monolingual resources for code-mixed Hindi-English speech recognition," in *IEEE International Conference on Asian Language Processing (IALP)*, 2017, pp. 218–221.

[24] S. Sivasankaran, B. M. L. Srivastava, S. Sitaram, K. Bali, and M. Choudhury, "Phone merging for code-switched speech recognition," in *Third Workshop on Computational Approaches to Linguistic Code-switching*, 2018.

[25] K. Taneja, S. Guha, P. Jyothi, and B. Abraham, "Exploiting monolingual speech corpora for code-mixed speech recognition," *in Interspeech*, pp. 2150–2154, 2019.

[26] S. Ganji, K. Dhawan, and R. Sinha, "IITG-HingCoS corpus: A Hinglish code-switching database for automatic speech recognition," *Speech Communication*, vol. 110, pp. 76–89, 2019.

[27] B. M. L. Srivastava, S. Sitaram, R. K. Mehta, K. D. Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, "Interspeech 2018 low resource automatic speech recognition challenge for Indian languages." in *SLTU*, 2018, pp. 11–14.

[28] *Indian Language TTS Consortium and ASR Consortium*, Indian Language Speech sound Label set (ILSL12), 2016, Online, Last accessed on 03-04-21.

[29] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[30] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI." in *Interspeech*, 2016, pp. 2751–2755.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[32] R. L. Weide, "The CMU pronouncing dictionary," *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*, 1998.

[33] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *International conference on spoken language processing*, 2002.