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

Anuj Diwan1, Rakesh Vaideeswaran2, Sanket Shah3, Ankita Singh1, Srinivasa Raghavan4, Shrey Khare5, Vinut Unni6, Saurabh Vyas1, Akash Rajpuria1, Chiranjeevi Yarra6, Ashish Mittal6, Prasanta Kumar Ghosh2, Preethi Jyothi1, Kalika Bali3, Vivek Seshadri3, Sunayana Sitaram3, Samarth Bharadwaj5, Jai Nanavati1, Raoul Nanavati1, Karthik Sankaranarayanan5,

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

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 ~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 recipe1 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, 21. This is similarly true for code-switched speech, and prior work has predominately focused on Hindi-English 22, 23, 24, 25, 26. MUCS 2021 challenge significantly contributes in this context, as we provide a larger corpus (~600 hours of transcribed speech from...
Table 1: Details of multilingual ASR train (Trn), test (Tst) and blind test (Blind) 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.

<table>
<thead>
<tr>
<th>Language</th>
<th>Hindi</th>
<th>Marathi</th>
<th>Odia</th>
<th>Telugu</th>
<th>Tamil</th>
<th>Gujarati</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size (hrs)</td>
<td>93.05</td>
<td>5.55</td>
<td>5.49</td>
<td>93.89</td>
<td>5</td>
<td>0.67</td>
</tr>
<tr>
<td>Ch.comp</td>
<td>PCM</td>
<td>PCM</td>
<td>PCM</td>
<td>PCM</td>
<td>PCM</td>
<td>PCM</td>
</tr>
<tr>
<td>Uniq sent</td>
<td>4506</td>
<td>386</td>
<td>316</td>
<td>2544</td>
<td>200</td>
<td>120</td>
</tr>
<tr>
<td>Spkrs</td>
<td>39</td>
<td>19</td>
<td>18</td>
<td>31</td>
<td>31</td>
<td>-</td>
</tr>
<tr>
<td>Vocab (words)</td>
<td>6092</td>
<td>1081</td>
<td>1339</td>
<td>5247</td>
<td>947</td>
<td>330</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Language</th>
<th>Hindi-Eng</th>
<th>Ben-Eng</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size (hrs)</td>
<td>89.86</td>
<td>5.18</td>
</tr>
<tr>
<td>Uniq sent</td>
<td>44249</td>
<td>2890</td>
</tr>
<tr>
<td>Spkrs</td>
<td>520</td>
<td>30</td>
</tr>
<tr>
<td>Vocab (words)</td>
<td>17830</td>
<td>3212</td>
</tr>
</tbody>
</table>

Different data sets were used to derive segments from the audio files to be aligned with the transcripts given in the text file. Table 2 shows the details of the data considered for Subtask2. 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.

2. Details of the two Subtasks

2.1. Subtask1: Multilingual ASR

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 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.

2.2. Subtask2: Code-switching ASR

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

2.1.3. 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).

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.
tion 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, 3) punctuation’s enunciation in the speech, 4) language mixing within a word, 5) incomplete audio at the beginning 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](https://github.com/Kyubyong/g2p)).

#### 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 respective segment.

Following this, we transliterate every English word in the reference transcription 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:** 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 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 

| 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 pairs. 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. This package provides pronunciations for keywords, retrieves pronunciations from CMUdict dictionary 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.

#### 3.2. Baseline results

##### 3.2.1. Multilingual ASR

Table 5 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 5 shows the amount of code-mix across the languages by averaging the percentage of words per sentence for a
language in the column in the decoded output of the utterances belonging to a 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 4 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 Blind, 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.

### 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 ~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]


