First Workshop on Speech Processing for Code-switching in Multilingual Communities: Shared Task on Code-switched Spoken Language Identification

Sanket Shah[1], *Sunayana Sitaram*[1], *Rupeshkumar Mehta*[2]

[1]Microsoft Research India
[2]Microsoft Corporation

{t-sansha, sunayana.sitaram, rupesh.mehta}@microsoft.com

Abstract

Code-switched speech and language processing is challenging due to the paucity of publicly-available datasets for research. We describe a shared task on language identification from speech organized as part of the First Workshop on Speech Technologies for Code-switching in Multilingual Communities. The shared task consisted two sub-tasks 1. Spoken Language Identification at the utterance level, where the goal was to classify an utterance as monolingual or code-switched 2. Spoken Language Identification within a code-switched utterance, where the goal was to identify languages at a frame-level. We released a dataset consisting of 60 hours of data in three language pairs - Tamil-English, Telugu-English and Gujarati-English. Six teams participated in the utterance level task, while two teams participated in the frame-level task. Team VocapiaLIMSI used a combination of i-vector and phonotactic-based models to achieve the best performance across languages on both tasks. We hope that this dataset, which is now available for research purposes will encourage research in code-switched spoken language identification.

Index Terms: spoken language identification, code-switching, shared task

1. Introduction

Code-switching, which is the use of two or more languages in a single conversation or utterance, is a challenging phenomenon for Speech and Natural Language Processing systems to handle. Recently, there has been significant progress made in code-switching Speech and NLP research [1], however, the lack of sufficient data and resources in code-switched language pairs compared to monolingual resources still remains a hurdle.

A well known technique for handling code-switching is to first identify the language of part of an utterance, and then use monolingual systems to process each corresponding fragment. Such Language Identification (LID) systems have also been used in conjunction with monolingual or multilingual systems to aid in processing. So far, the focus of code-switching research has mainly been on LID in text, in which words in a code-switched utterance are individually labeled with language tags. Much of this research has been spurred due to several shared tasks on code-switched text LID. Inspired by this, we conducted a shared task on Language Identification from speech, which has been relatively less studied.

Our shared task consisted of two sub-tasks. Task A was an utterance-level LID task, in which a system had to classify an utterance as being monolingual or code-switched. Task B was a frame-level LID task, in which a system had to classify a code-switched utterance with language labels at the frame-level. For both tasks, systems only had access to raw audio at test time, without accompanying text transcriptions of the utterances.

For the shared task, we released a code-switched speech corpus with 60 hours of speech from three language pairs: Tamil-English, Telugu-English and Gujarati-English. This dataset is available for research use and is the first dataset of code-switched speech in these language pairs. More details about the dataset can be found in Section 4. We used a CNN-BLSTM-based model to build baselines for both tasks, which is described in Section 5. Participants were provided with a blind test set to evaluate their systems on. We describe the techniques used by participating teams in the shared task, along with results in Section 6.

2. Related work

Language Identification for code-switched utterances from text has been well studied and a comprehensive overview can be found in [1]. The first and second workshops on Computational Approaches to Code Switching conducted a shared task on Language Identification for several language pairs [2, 3].

While LID for speech has been a well established area of research, intra-utterance LID from speech for code-switching has been relatively less studied. However, there have been initial attempts to solve this problem, mainly in the context of using LID systems in conjunction with an Automatic Speech Recognition (ASR) system.

In [4], the authors show that humans exploit prosodic cues to detect code switching in speech and can anticipate switch points even in noisy speech. In [5, 6] the authors investigate the effectiveness of using retrained multilingual DNNs and augmenting the data for detecting the language. In [7, 8] authors employ word based lexical information, build HMM-based acoustic models followed by an SVM based decision classifier to identify code-switching between Northern Sotho and English [9]. It may be useful for an ASR system to be able to detect the code-switching style of a particular utterance, and be able to adapt to that style through specialized language models or other adaptation techniques. [10] classify code-switched corpora by code-switching style and show that features extracted from acoustics alone can distinguish between different kinds of code-switching in a single language.

Previously, we conducted a low resource ASR challenge [11] in which we released 150 hours of speech data for three languages: Tamil, Telugu and English. In this challenge, we release code-switched data for the same language pairs so that the research community can make use of this data together to build robust speech systems for multilingual scenarios.

3. Challenge Rules

As mentioned before, the shared task consisted of two subtasks:

- Task A: Utterance-level identification of monolingual vs.

code-switched utterances

- Task B: Frame-level identification of language in a code-switched utterance.

Registered participants were sent a link to download data sets in all three languages. We released 16 hours of training data and around 2 hours of test data for each language, details of which can be seen in section 4. We also released baseline Accuracy (ACC) and Equal Error Rate (EER) numbers evaluated on the 2 hour test sets along with instruction on how to replicate the baselines. Details of the baseline and evaluation metrics can be found in section 5. Participants were allowed to use monolingual data that we released in our previous ASR challenge [12], but no other external data was allowed to be used.

Testing was conducted in April 2020, in which we tested all submitted systems over a period of 3 days. Participants were sent links to 4 hours of blind test audio data in each language for Task A and 2 hours of blind test audio data in each language for Task B. Participants ran their systems on the blind test audio data and submitted hypothesis files via email to an automated scoring system which calculated the ACC and EER and sent it back to them as a reply to their email. Each participating team was allowed 3 attempts per language per task, so each team could submit up to 18 models in all for evaluation. The automated scoring system was created using Microsoft Flow [1].

4. Data

4.1. Data

The data released for the challenge was provided by SpeechOcean.com and Microsoft. It consisted of phrasal (recorded as read-out phrases) and conversational speech in Tamil-English, Telugu-English and Gujarati-English. The transcription consisted of English words in Roman script and the other language's words in native script, although there was some cases of cross-transcription. Some example code-switched utterances are shown in figure 1.

4.1.1. Task A

For training data, each character in the transcript was replaced its corresponding language tag by looking up the language it was transcribed in i.e., 'T' for Telugu or Tamil and 'G' for Gujarati. As a part of training data, participants were allowed to use data from Low Resource Automatic Speech Recognition Challenge [11] along with the data released as a part of this challenge. The test and blind test sets consisted of monolingual and code-switched utterances and their corresponding class (0 and 1 respectively). The data description for task A can be seen in table 1.

Table 1: *Description of train, test and blind test data for Task A*

	Train	***Test***	***Blind Test***
ta-en	16 hrs (8928 utt.) (CMI:0.17)	4 hrs (2223 utt.)	4 hrs (2235 utt.)
te-en	16 hrs (8226 utt.) (CMI:0.22)	4 hrs (2149 utt.)	4 hrs (2133 utt.)
gu-en	16 hrs (8620 utt.) (CMI:0.16)	4 hrs (2091 utt.)	4 hrs (2102 utt.)

The Code Mixing Index [13] is a metric that measures the amount of code-switching in a corpus by using word frequencies. We report the CMI of our code-switched train and test sets in parentheses in Table 1 and Table 2. Gujarati and Tamil have a relatively lower CMI of 17% while the Telugu dataset has a CMI of 22%.

4.1.2. Task B

Since Task A was an utterance-level task, each utterance had a single label. In task B, each utterance was labeled with language information at the frame-level. In the training data, each character in the transcript was replaced its corresponding language tag i.e., 'T' for Telugu or Tamil and 'G' for Gujarati. For the test and blind test sets, we generated language tags for every 200 ms of the CS audio. Further details of the data for task B can be found in table 2.

Table 2: *Description of train, test and blind test data for Task B*

	Train	***Test***	***Blind Test***
ta-en	16 hrs (8928 utt.) (CMI:0.17)	2 hrs (1135 utt.) (CMI:0.18)	2 hrs (1120 utt.) (CMI:0.17)
te-en	16 hrs (8226 utt.) (CMI:0.22)	2 hrs (1047 utt.) (CMI:0.22)	2 hrs (1033 utt.) (CMI:0.22)
gu-en	16 hrs (8620 utt.) (CMI:0.16)	2 hrs (1080 utt.) (CMI:0.17)	2 hrs (1078 utt.) (CMI:0.17)

5. Evaluation and Baselines

5.1. Evaluation

We used two standard metrics for evaluation: Accuracy (ACC) and Equal Error Rate (EER). They are defined as follows.

- Accuracy

$$Accuracy = \frac{N}{T}$$

Where,
N is the total no. of correctly predicted data samples
T is the total no. of data points

- Equal Error Rate (EER)

$$EER = \frac{FRR + FAR}{2}$$

$$FRR = \frac{TFR}{T}$$

$$FAR = \frac{TFA}{T}$$

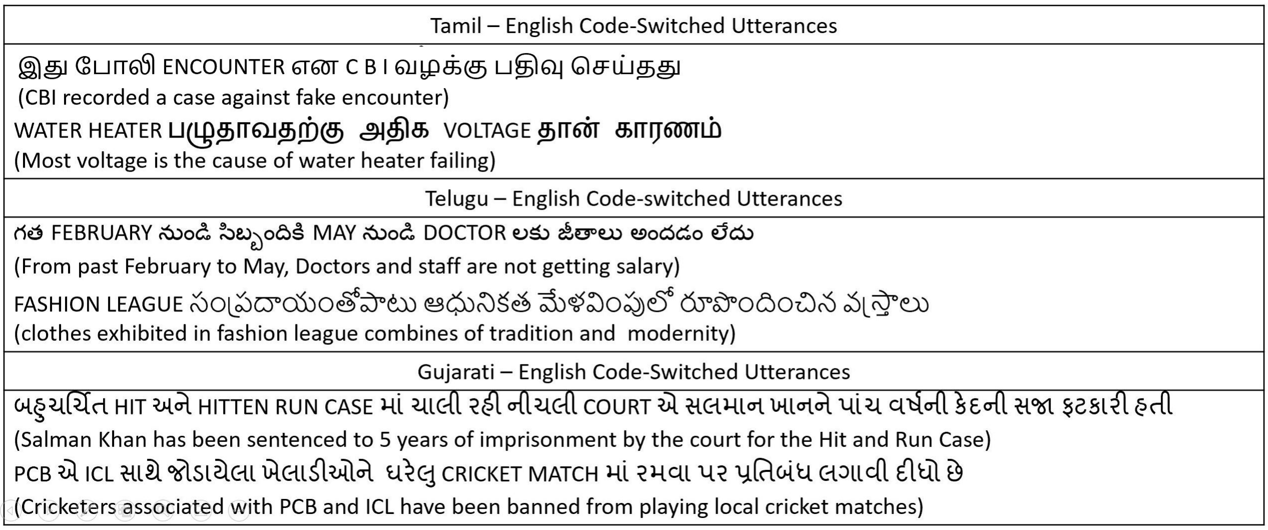
where,
EER Equal Error Rate
FRR False Rejection Rate
FAR False Acceptance Rate
TFR Total No. of False Rejects
TFA Total No. of False Accepts
T Total No. of Datapoints

5.2. Baselines

We used a CNN-BLSTM model based on Deepspeech2 [14] as the baseline system for both tasks.

We denote our training monolingual datasets (X_1^L, Y_1^L),...,(X_n^L, Y_n^L) where $L \in$ {TE/TA/GU}, code-switched

[1] https://flow.microsoft.com/



Tamil – English Code-Switched Utterances
இது போலி ENCOUNTER என C B I வழக்கு பதிவு செய்தது (CBI recorded a case against fake encounter) WATER HEATER பழுதாவதற்கு அதிக VOLTAGE தான் காரணம் (Most voltage is the cause of water heater failing)
Telugu – English Code-switched Utterances
గత FEBRUARY నుండి సిబ్బందికి MAY నుండి DOCTOR లకు జీతాలు అందడం లేదు (From past February to May, Doctors and staff are not getting salary) FASHION LEAGUE సంప్రదాయంతోపాటు ఆధునికత మేళవింపులో రూపొందించిన వస్త్రాలు (clothes exhibited in fashion league combines of tradition and modernity)
Gujarati – English Code-Switched Utterances
બહુચર્ચિત HIT અને HITTEN RUN CASE માં ચાલી રહી નીચલી COURT એ સલમાન ખાનને પાંચ વર્ષની કેદની સજા ફટકારી હતી (Salman Khan has been sentenced to 5 years of imprisonment by the court for the Hit and Run Case) PCB એ ICL સાથે જોડાયેલા ખેલાડીઓને ઘરેલુ CRICKET MATCH માં રમવા પર પ્રતિબંધ લગાવી દીધો છે (Cricketers associated with PCB and ICL have been banned from playing local cricket matches)

Figure 1: *Some example code-switched utterances from the dataset.*

datasets (X_1^{CS}, Y_1^{CS}),...,(X_n^{CS}, Y_n^{CS}) where $CS \in$ {TE-EN/TA-EN/GU-EN}. The labels Y are language tags i.e., 'T' for Telugu or Tamil and 'G' for Gujarati. Our baseline model consists of two Convolution Neural Network (CNN) layers followed by five bidirectional long-short term (BLSTM) layers of 1024 dimension. Further, the frame-wise posterior distribution $P(Y|X)$ is conditioned on the input X and calculated by applying a fully-connected layer and a softmax function.

$$P(Y|X) = \text{Softmax}(Linear(h)) \tag{1}$$

where h is the hidden state from BLSTM. For task A, based on the softmax output, we determine whether the audio is monolingual or code-switched, while for task B, We use the 200ms frame length and predict the language tag for the same. The model parameters are trained using the Connectionist Temporal Classification (CTC) [15] criterion. We use the SGD optimizer with 3e-4 learning rate. We trained the model for 40 epochs for Task A and 60 epochs for Task B, with mini-batch size equal to 64 per GPU.

Baseline results for task A and task B are provided in table 3 and 4 respectively.

Table 3: *Baseline results for Task A*

Class 1	***Class 2***	***ACC***	***EER***
ta-en	ta-mono	74.0%	13.0%
te-en	te-mono	71.2%	14.4%
gu-en	gu-mono	76.8%	11.6%

Table 4: *Baseline results for Task B*

	ACC	***EER***
ta-en	77.6%	6.5%
te-en	76.5%	6.7%
gu-en	76.7%	6.7%

6. Systems and Results

Six teams participated in task A, while two teams participated in task B. In this section, we describe the systems built by participating teams and compare their results to the baseline numbers. The teams which participated in the evaluation were VocapiaLIMSI, Swiggy, AnnotateIt, Sizzle, Ground Zero and CMU. Participants spanned academic institutions, industry labs and startups.

Tables 5 and 6 show results from participating teams on both tasks. For task A, in case of Gujarati, no team was able to beat the baseline in terms of ACC and EER. In case of Telugu, four teams (VocapiaLIMSI, Swiggy, CMU and Sizzle) outperformed the baseline. In case of Tamil, two teams (VocapiaLIMSI and Swiggy) outperformed the baseline. For task B, the VocapiaLIMSI system outperformed the baseline for all three language pairs, while the Swiggy team came close to but could not outperform the baseline. Next, we describe the approaches that the various teams took for solving both tasks.

Team VocapiaLIMSI [16] achieved the best scores for both the tasks, and across all the languages. They proposed two acoustic modeling approaches, one being i-vector modeling of the audio segments and phonotactic modeling focusing on sequences of language-independent phone units. The i-vector system characterizes the language of an utterance with vectors obtained by projecting the speech data onto a total variability space while the phonotactic approach relies on the idea that the phonetic sequence in an audio sample is characteristic of the language used. Team VocapiaLIMSI [16] also experimented with combining the scores from both the systems and found that a linear combination of the scores from both the systems gave the best performance.

Team Swiggy [17] modified the spectral augmentation approach and proposed a language mask that discriminated language ID pairs, leading to a noise robust spoken LID system. The authors proposed a temporal masking approach in which they first map language transcripts to speech frames. The number of temporal masks were determined based on the number of English words present in the language transcripts. Time segments corresponding to the English words were masked. Once the masking was done, they then extracted features and passed

Table 5: *ACCs and EERs for top performing models for task A*

	Team	*ACC*	*EER*
tamil	Baseline	74.0%	13.0%
tamil	VocapiaLIMSI	79.8%	10.1%
tamil	Swiggy	78.6%	10.6%
tamil	CMU	73.6%	13.2%
tamil	Sizzle	69.1%	15.5%
tamil	Ground Zero	67.1%	17.0%
tamil	AnnotateIt	61.8%	19.1%
telugu	Baseline	71.2%	14.4%
telugu	VocapiaLIMSI	79.3%	10.3%
telugu	Swiggy	79.0 %	10.5%
telugu	CMU	73.9%	13.0%
telugu	Sizzle	71.3%	14.3%
telugu	Ground Zero	67.2%	16.4%
telugu	AnnotateIt	59.6%	20.2%
gujarati	Baseline	76.8%	11.6%
gujarati	VocapiaLIMSI	75.3%	12.3%
gujarati	Swiggy	73.0%	13.5%
gujarati	AnnotateIt	66.1%	16.9%
gujarati	Ground Zero	55.1%	22.4%
gujarati	CMU	49.9%	25.0%
gujarati	Sizzle	48.4%	25.7%

Table 6: *ACCs and EERs for top performing models for task B*

	Team	*ACC*	*EER*
tamil	Baseline	77.6%	6.5%
tamil	VocapiaLIMSI	78.8%	6.5%
tamil	Swiggy	76.1%	7.4%
telugu	Baseline	76.5%	6.7%
telugu	VocapiaLIMSI	79.6%	6.3%
telugu	Swiggy	76.1%	7.4%
gujarati	Baseline	76.7%	6.7%
gujarati	VocapiaLIMSI	77.7%	6.9%
gujarati	Swiggy	76.1%	7.5%

them to an end-to-end CNN-LSTM system and train the system using CTC loss function at the output layer.

Team CMU [18] proposed a multi-task learning framework where the primary task is language detection and secondary task is audio reconstruction. Considering a speech corpus X consisting of languages $\{l_1, .., l_n\}$, where each l_i comprise of difference speakers. $x_1, ..., x_n$ denotes acoustic frames of X. x_i can be monolingual or code-switched and Y denotes the labels. The authors trained a model to learn the join distribution between $\{x, y\}$. The process was mediated using latent discrete random representation. To ensure that the latent representations correspond to speech utterances, they adopted this multi-task learning framework.

Team AnnotateIt [19] used a two-stage training and inference model. In the front-end, the authors trained a monolingual speech recognition model using the monolingual speech corpus described in the previous section. In the second stage, they used the inference from the first stage to train a linear classifier for the binary language identification task: monolingual vs. code-switched.

Team Sizzle [20] proposed a convolutional encoder in combination with the transformer architecture for utterance-level code-switching detection. The authors used the convolutional encoder to process audio features and pass these features to a transformer based network with multi-head self attention layers. Finally, the authors, aggregated the frame level features from the transformer network to get utterance level features to predict the class label.

In summary, approaches that used the audio directly and did not rely on an ASR system seemed to work best, however this may be due to the small amount of data used to train the ASR. Future work in this direction could include using all the languages in a multilingual setup for spoken language identification.

7. Conclusions

In this paper, we described the first shared task conducted for spoken language identification of code-switched speech. The shared task consisted of two sub-tasks - an utterance-level task to classify speech into monolingual or code-switched and an intra-utterance classification task, in which a code-switched utterance had to labeled with languages at the frame-level. We released, for the first time, code-switched data in three language pairs: Tamil-English, Telugu-English and Gujarati-English for the shared task. The data released as part of this shared task is available for future research use.

Six teams participated in the shared task, however we had significantly less participation in task B compared to task A. The best performing systems that beat the baseline included VocapiaLIMSI and Swiggy, which used approaches based on i-vectors, phonotactic models and temporal masking. The second task of our shared task, which is the frame-level identification of code-switching remains under-explored, and we hope that future research will focus on this problem.

8. Acknowledgements

The authors would like to thank Sandeepkumar Satpal and Vinay Krishna from Microsoft for help with creating the data for the shared task.

9. References

[1] 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.

[2] T. Solorio, E. Blair, S. Maharjan, S. Bethard, M. Diab, M. Ghoneim, A. Hawwari, F. AlGhamdi, J. Hirschberg, A. Chang *et al.*, "Overview for the first shared task on language identification in code-switched data," in *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 2014, pp. 62–72.

[3] G. Molina, F. AlGhamdi, M. Ghoneim, A. Hawwari, N. Rey-Villamizar, M. Diab, and T. Solorio, "Overview for the second shared task on language identification in code-switched data," *arXiv preprint arXiv:1909.13016*, 2019.

[4] P. E. Piccinini and M. Garellek, "Prosodic cues to monolingual versus code-switching sentences in english and spanish," in *Proceedings of the 7th Speech Prosody Conference*, 2014, pp. 885–889.

[5] E. Yılmaz, H. van den Heuvel, and D. van Leeuwen, "Code-switching detection using multilingual dnns," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 610–616.

[6] E. Yılmaz, H. v. d. Heuvel, and D. A. van Leeuwen, "Code-switching detection with data-augmented acoustic and language models," *arXiv preprint arXiv:1808.00521*, 2018.

[7] D.-C. Lyu, R.-Y. Lyu, C.-L. Zhu, and M.-T. Ko, "Language identification in code-switching speech using word-based lexical model," in *2010 7th International Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 460–464.

[8] D.-C. Lyu, T.-P. Tan, E.-S. Chng, and H. Li, "An analysis of a mandarin-english code-switching speech corpus: Seame," *Age*, vol. 21, pp. 25–8, 2010.

[9] K. R. Mabokela, M. J. Manamela, and M. Manaileng, "Modeling code-switching speech on under-resourced languages for language identification," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[10] S. Rallabandi, S. Sitaram, and A. W. Black, "Automatic detection of code-switching style from acoustics," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, 2018, pp. 76–81.

[11] B. M. L. Srivastava, S. Sitaram, K. Bali, R. K. Mehta, K. D. Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. N. , "Interspeech 2018 low resource automatic speech recognition challenge for indian languages," in *SLTU*, August 2018. [Online]. Available: https://www.microsoft.com/en-us/research/publication/interspeech-2018-low-resource-automatic-speech-recognition-challenge-for-indian-languages/

[12] B. M. L. Srivastava and S. Sitaram, "Homophone Identification and Merging for Code-switched Speech Recognition," *Proeedings of Interspeech 2018*, 2018.

[13] B. Gambäck and A. Das, "On measuring the complexity of code-mixing," in *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*, 2014, pp. 1–7.

[14] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 173–182.

[15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[16] J.-L. G. Claude Barras, Viet-Bac Le, "Vocapia-limsi system for 2020 shared task on code-switched spoken language identification," 2020.

[17] H. M. Pradeep Rangan, Sundeep Teki, "Exploiting spectral augmentation for code-switched spoken language identification," 2020.

[18] A. W. B. Sai Krishna Rallabandi, "On detecting code mixing in speech using discrete latent representations," 2020.

[19] J. A. C. Parav Nagarsheth, "Language identification for code-mixed indian languages in the wild," 2020.

[20] A. P. Krishna D N, "Utterance-level code-switching identification using transformer network," 2020.