Language Identification for Code-Mixed Indian Languages In The Wild

Parav Nagarsheth, Jehoshaph Akshay Chandran

AnnotateIt

parav@annotateit.com, akshay@annotateit.com

Abstract

In this paper, we identify challenges for language identification for code-mixed Indian languages with speech utterances captured in the wild. The study begins with building models for controlled datasets for Telugu, Tamil and Gujarati code mixed speech. This is part of the shared task of the Speech Technologies for Code-Switching in Multilingual Communities 2020 workshop. In the second half of the paper, we discuss an exercise to collect more diverse code-switched data for evaluating the language identification models. This data is derived from a wide range of YouTube videos in Gujarati and Telugu. The models get mixed results: the development set shows excellent performance, while the evaluations show degradation. The study identifies a need for more diverse, real-world code-mixed datasets for underserved languages.

Index Terms: multilingual speech recognition, code-mixed speech, language identification

1. Introduction

One of the challenges of building speech recognition models for modern Indian languages is the use of multiple languages in the same utterance. Unlike monolingual systems, multilingual or code-mixed systems need to be able to deal with utterances or words in different languages. Moreover, many publicly available datasets are imbalanced [1], with an over-representation of the primary language (or mother-tongue) of the speaker, or an over-reliance on written text to derive spoken utterances. This motivates our call for building higher quality datasets for code-mixed speech and NLP tasks.

To mitigate some of these challenges, some systems have explored conditioning the speech recognition models with a language vector [1, 2, 3, 4]. This method has shown to improve speech recognition performance for large multilingual speech recognition models [1]. In this paper, we study the task of language identification for code-mixed languages for use in downstream speech recognition tasks.

Our approach first builds monolingual speech models for the three languages (Telugu, Tamil, and Gujarati), which are subsequently used to train a binary classifier to predict between code-mixed and monolingual utterances. Our approach also takes into account the quantity and quality constraints of the available training data.

To build monolingual speech models, we use a modified Wav2Letter model [5] to train a simple speech recognition model. For the binary classification task, we use a linear SVM that we pass the character n-grams of the monolingual speech model.

In the second part of the paper, we present a framework for the collection of a large volume of high quality and well-annotated modern vernacular speech in realistic settings. We build a high-quality code-mixed dataset in two steps. Step one involves the collection of diverse content from YouTube consisting of interviews, educational videos, local film industry inspired by earlier work that used YouTube celebrity interviews to curate a vast dataset of English language speakers [14]. We use the SyncNet framework to extract audio clips from these videos. In the next step, to annotate this data, we build an annotation tool to annotate multilingual speech datasets. The tool uses several techniques to make annotating audio clips easy using tools inspired by the state of the art research in annotation of code-mixed speech.

Finally, we evaluate the performance of the language identification model on the shared task from the Code-Switching in Multilingual Communities 2020 Workshop and the YouTube dataset and analyze the underperformance of the model for the latter.

2. Related Work

Considerable effort has gone into building datasets and models for code-mixed and multilingual speech.

2.1. Speech Recognition and Language Identification Models

Anjuli Kannan et. al. [1] propose a low latency end-to-end system that works with real world data where they use a single multilingual model instead of separate models for each language. They propose sampling as one method to reduce the problem with imbalance in multilingual utterances. Additionally, they found the use of language vectors useful in improving the modified transliterated Word Error Rates (WER) [6]. In a review of the Shared Task Evaluation for Language Identification at VarDial, several top performing submissions were found to use linear discriminators for language identification [7]. Feature hashing has been found helpful to reduce the high-dimensionality of language identification datasets [8].

2.2. Data Annotation

Sanket Shah et. al. [9] describe an approach to building an annotation tool for collecting code switched data. Their annotation interface helps annotators transcribe code-switched speech faster and more accurately than a traditional input tool. In later sections, we use a similar interface for annotation, with some modifications to accommodate for input on mobile devices.

3. Shared Task

The Workshop on Speech Technologies for Code-switching in Multilingual Communities 2020 organized a shared task for Code-switched Spoken Language Identification (LID) for these language pairs:

1. **Gujarati-English** (GU-EN)
2. **Telugu-English** (TE-EN)
3. **Tamil-English** (TA-EN)

The shared task has two parts:

- **Part A**: Language identification at the utterance level
- **Part B**: Language identification at the frame level

For each of the sub-parts, datasets have speech data labeled for code-mixed and monolingual at the utterance and frame-level respectively.

The language identification models discussed in the following section solve the problem of utterance level language identification (Part A), which allows utterance level language vector conditioning for downstream speech recognition models.

3.1. Dataset Description

For Part A, the dataset for each language has been split into a training dataset and a development dataset. Each utterance is between 2 seconds to 24 seconds long, and the utterance is labeled as either monolingual or code-mixed speech.

The provenance of the data was not available from the organizers at the time of publication. However, after sampling a few audio files, it seems likely that the dataset is speech recorded by narrating target language text corpora. Due to constraints posed by recording in these conditions, the dataset may not cover certain real-world domains like spontaneous speech, colloquial speech and conversations.

Additionally, a much larger collection of monolingual data [1] was available for each target Indian language with utterance labels, that could be used to train the language identification model, with the condition that test data from this dataset should be excluded.

4. Language Identification Model

For language identification, we use a two-stage training and inference model. In the front-end, we train a monolingual speech recognition model using the monolingual speech corpus described in the previous section. In the second stage, we use the inference from the first stage to train a linear classifier for the binary language identification task: monolingual vs. code-mixed.

4.1. Monolingual Speech Recognition Model

Since CNNs are generally faster than LSTMs, we use a Wav2Letter model with CTC [10] loss to train a simple speech recognition model from the monolingual data.

The monolingual data has unicode sentence labels for Gujarati, Tamil and Telugu speech. Gujarati uses a variant of the Devanagri script, while Tamil and Telugu come from the family of Brahmic scripts. These languages have high grapheme-to-phoneme correspondence, hence grapheme based models should theoretically achieve similar performance to phoneme based models, even for relatively small amounts of training data. Moreover, there is very little phonetic data available for these languages.

End-to-end speech recognition models trained on character-based outputs like graphemes, byte-pair-encodings (BPEs), or word-pieces, jointly learn the acoustic model, pronunciation model and language model within a single neural-network [11, 12]. The latter two would be particularly useful during inference, since code-mixed speech with English words will be effectively treated as out-of-vocabulary (OOV) words.

[1] https://msropendata.com/datasets/7230b4b1-912d-400e-be58-f84e0512985e

We use 90-dimensional log-mel spectrum with a 20 millisecond window and a 10 millisecond hop length as frame level audio features. For the output, we tokenize the sentences using BPE with a vocabulary of size 1000.

The monolingual speech corpus is split into two parts: 80% of the corpus is used for fitting the model parameters using backpropagation and 20% of the corpus is used to keep track of the validation character error rate (CER). We also use audio augmentation (pitch shift and speed change) and spectral augmentation [13] to improve generalization performance.

To avoid overfitting, no additional hyperparameter search is performed. Additionally, to encourage better generalization performance, we use the same hyperparameters for training monolingual speech recognition models of all three languages.

The best performing speech models trained above are generally not suitable for practical speech recognition applications, however, the output is discernible enough to train downstream discriminators as described in the next section.

4.2. Language Identification Binary Task

From the tokenized output of the monolingual speech model, we extract character n-grams of length 1 to 5. For training a linear discriminator, these n-grams need to be categorized. However, because of the large space of possible n-grams, the dimensionality of the categorized features is very high.

Feature hashing provides an elegant way of reducing dimensionality without limiting the n-gram space. We use 1000 dimensional feature hashing to reduce the dimensionality of the Bag-of-Words (BoW) n-gram vectors for input to the classifier.

The features are used as input to a linear SVM that predicts between code-switched and monolingual utterances.

5. YouTube Data Collection and Annotation

The availability of high-quality resources for low-resource language is a significant constraint in building speech and natural language models. Each of India's top-10 native languages have more than 30 million speakers each. However, the amount of labeled speech data available is minuscule in comparison to other languages like English, Mandarin or Russian.

Since Indians are naturally multilingual, informal speech and text includes frequent code-mixing to a high-degree. On the other hand, formal text and speech, often used for training speech and language models, contains very little code-mixing.

With the proliferation of low-cost high-speed internet, there is a sizable amount of Indian language content available online on social media, video sharing platforms and streaming platforms. YouTube, in particular, hosts a large number of Indian content creators that cater diverse interests including celebrity interviews, news, educational videos, Bollywood film and music, fashion and beauty tips, agriculture, stand-up comedy, pranks etc.

Earlier work had used YouTube celebrity interviews to curate a vast dataset of English language speakers for speaker recognition applications [14]. Inspired from that work, we use a similar workflow to curate a small dataset of Gujarati and Telugu speech from YouTube videos for speech recognition.

5.1. Gujarati and Telugu Speech from YouTube Videos

We use the SyncNet framework [15] to extract speech from YouTube videos. The SyncNet model uses face tracking and lip tracking to sync speech with the speaker in a video frame. For the interested readers, we refer them to the original paper.

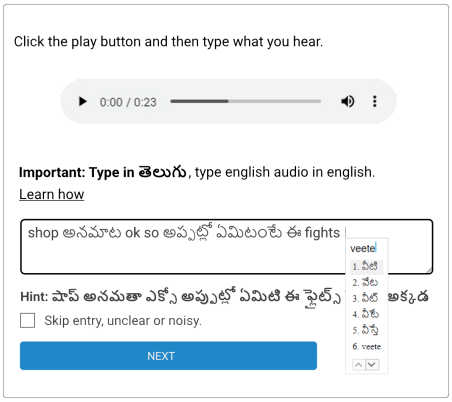

Figure 1: *Code-Mixed Speech Annotation Interface*

From the output of the model, audio clips are segmented into utterances of up to 30 seconds length to allow batch training of neural networks.

For the source Gujarati and Telugu videos, we choose topics that cover local celebrity interviews, official government bulletins, current affairs, food preparation videos and educational videos, where the speaker is visible in the frame most of the time. Videos with music are filtered out. The common theme among all of the videos is that they have a large amount of code-mixing with English.

5.2. Speech Annotation using Crowdsourcing

Crowdsourcing is commonly used to get speech annotations for low-resource languages. Crowdsourcing can be cheap, however quality can be a concern when annotators are untrained and focused on maximizing their throughput. For annotating the speech utterances derived from YouTube videos, we created a custom interface for our annotators, as shown in Figure 1. The annotators are able to playback the speech utterance on a desktop or mobile platform. Annotators can quickly switch between a standard English keyboard to type English words or a transliteration keyboard to type Gujarati/Telugu words. They are, optionally, given a transcription hint to increase annotation throughput.

For quality control we take two measures:

- For every annotation session, we benchmark annotator accuracy using a pool of expert-labeled speech, and reject utterances where the annotator accuracy falls below a calibrated quality threshold.
- For further reliability, we merge redundant annotations for the same utterance with a consensus algorithm similar to ROVER [16]. Each evaluation utterance has roughly 3-5 redundant annotations.

Table 1: *Number of utterances of code-mixed and monolingual speech in the YouTube dataset*

	Number of Utterances	
	Monolingual	**Code-Mixed**
Gujarati	766	517
Telugu	697	754

Table 2: *Results of Gujarati, Telugu and Tamil code-mixed identification (Part A of shared task evaluation)*

	Organizer Baseline		**Our Submission**			
	Dev		**Dev**		**Eval**	
	Acc.	**EER**	**Acc.**	**EER**	**Acc.**	**EER**
GU	76.8%	11.6%	81.2%	9.4%	66.1%	16.9%
TE	71.2%	14.4%	85.9%	7.1%	59.7%	20.1%
TA	74.0%	13.0%	85.6%	7.2%	61.8%	19.1%

To identify code-mixed speech, we sample the utterances for the presence of one or more words in the Latin script. This is not always accurate, for example in Gujarati, it is common to find English words suffixed with Gujarati morphemes to express the plural:

- Case becomes કેસો instead of cases
- Film becomes ફિલ્મો instead of films

Moreover, Indian languages use a lot of loaner words from English, which are interchangeably written in Latin or regional scripts. To minimize errors from these peculiarities, we reject utterances where there is annotator disagreement about specific words in the consensus building step.

In this exercise, we collected over 200 hours of annotated Gujarati and Telugu speech. However, for practical reasons, we chose a significantly smaller dataset for evaluation as shown in Table 1. The dataset is fairly balanced between code-mixed and monolingual utterances.

6. Results

6.1. Shared Task Evaluation

The results of the shared task evaluation are shown in Table 2. The language identification model outperforms the baseline on the development set for all three languages. This is encouraging, since the individual models do not have any language specific hyperparameter tuning.

There is a notable drop in accuracy and equal error rate (EER) for the testing set, as compared to the development set, yet the models perform comparably to the leaderboard available at the time of submission. The Gujarati language identification model, in particular, was ranked at third place in the leaderboard among five submissions hosted on the organizer's website. Due to a late submission of the evaluation, our results were not officially published on the leaderboard.

Labels for the test set were not available at the time of publishing for error analysis. However, we hypothesize that the development set for the code-mixed data and the training set for the monolingual corpus may have significant overlap for monolingual utterances. This may indicate that the model has not

Table 3: *Accuracy and EER for code-mixed language identification on YouTube dataset*

	YouTube	
	Accuracy	**EER**
Gujarati	42.1%	28.9%
Telugu	52.7%	23.6%

generalized to unseen monolingual speech data. Another hypothesis is that the provenance of the test dataset is significantly different from the training and development dataset.

6.2. Cross-Domain YouTube Dataset Evaluation

For the YouTube dataset, we generated ground truth labels from the annotated transcripts after data cleaning and quality control. The same Gujarati and Telugu models are used for evaluation as in the section 6.1, without any fine-tuning. Hence, this can classified as a cross-domain evaluation.

In Table 3, the performance of the language identification model trained on data from **Part A** of the Shared Task suffers a significant drop compared to the test results from the previous section. The model particularly underperforms for Gujarati, where we saw the best performance in the evaluation of the Shared Task among all three languages.

On further analysis, it was observed that monolingual utterances were frequently being erroneously classified as code-mixed. Since the binary classifier is presumably functioning as an anomaly detector, this may indicate that the front-end needs more in-domain monolingual data to improve generalization performance.

7. Conclusion and Future Work

In this submission we propose a two step approach to language identification for a code-switched utterance. We build a traditional monolingual speech recognition model using Wav2Letter. Next we use the n-grams of the monolingual model as input for a linear SVM to perform the binary classification task. From the data provided by the challenge, we show promising results on the dev set and peer-comparable results for the shared task evaluation. However, there is a notable degradation in accuracy and EER for the cross-domain YouTube dataset evaluation. The drop in accuracy could be ascribed to:

- Lack of large amounts of in-domain training data
- Lack of model generalization
- Noisy labels for monolingual and code-mixed speech

These results clearly demonstrate a pressing need for evaluating models on real-world and cross-domain datasets for speech recognition tasks in underserved languages. Datasets can be curated relatively inexpensively and scaled up quickly with the use of right annotation tools and quality control mechanisms.

This paper also recognizes the need for larger datasets for downstream applications in NLP for underserved languages. In the future, we anticipate participation in dataset annotation and collection exercises in text to speech, comprehension, intent detection and more for a wider range of Indian languages.

8. Acknowledgements

We are thankful to annotators who spent several hours annotating code-mixed Gujarati and Telugu speech for this project.

9. References

[1] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," *arXiv preprint arXiv:1909.05330*, 2019.

[2] 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 *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4904–4908.

[3] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4749–4753.

[4] M. Grace, M. Bastani, and E. Weinstein, "Occams adaptation: A comparison of interpolation of bases adaptation methods for multi-dialect acoustic modeling with lstms," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 174–181.

[5] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[6] J. Emond, B. Ramabhadran, B. Roark, P. Moreno, and M. Ma, "Transliteration based approaches to improve code-switched speech recognition performance," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 448–455.

[7] M. Zampieri, S. Malmasi, Y. Scherrer, T. Samardzic, F. Tyers, M. Silfverberg, N. Klyueva, T.-L. Pan, C.-R. Huang, R. T. Ionescu *et al.*, "A report on the third vardial evaluation campaign," in *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, 2019, pp. 1–16.

[8] S. Malmasi and M. Dras, "Feature hashing for language and dialect identification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 399–403.

[9] S. S. P. J. S. Santy and S. Sitaram, "Cossat: Code-switched speech annotation tool," *EMNLP 2019*, p. 48, 2019.

[10] 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*, 2006, pp. 369–376.

[11] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," *Proc. Interspeech 2019*, pp. 3800–3804, 2019.

[12] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[13] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[14] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Science and Language*, 2019.

[15] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Asian conference on computer vision*. Springer, 2016, pp. 251–263.

[16] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997, pp. 347–354.