Spoken Language Identification from Audio Data for Uncommon Indian Languages

Deepthi Raghu
draghu@iu.edu

Abstract

Language identification has become an important part of many speech recognition systems to classify the spoken language from audio data. With the introduction of many voice-based assistants, spoken language identification is becoming an important study area. Another usecase of this work is to tag uncommon languages spoken in YouTube videos. Tagging these uncommon languages will help in expanding the number of languages for which subtitles can be submitted on YouTube. This project involves identification of the uncommon languages spoken from audio data. The idea is to first translate the audio data into English text using an automatic speech recognizer. As the next step, an ngram model is trained on the transcribed English output to identify and tag the corresponding language. Uncommon Indian languages (Tamil, Odia, Assamese and Punjabi) from Mozilla's Common Voice dataset are used as training data for the language identification model. To evaluate the model, audios downloaded from YouTube are used as evaluation/test data. An accuracy of 50.67% was obtained from the evaluation results.

1 Introduction

Many intelligent voice assistants such as Siri, Alexa and Google Assistant use speech recognition in their system. Captioning mechanisms in platforms like YouTube also use speech recognition to provide accurate captions to videos. In these systems, language identification plays a major role to automate the speech recognition process. This paper describes a language identification system for uncommon Indian languages Tamil, Odiya, Assamese and Punjabi. Although these languages are less popular outside India, there are many people in India who watch videos on YouTube or use voice assistants in this language. Finding effective methods to identify these languages will be very helpful in accurate speech recognition or caption generation in these languages. In this paper, audio dataset from Mozilla's Common Voice is used as the training data. For the testing data, audio samples from YouTube has been collected. There are two parts involved in this language identification system. First, the audio data is fed into an automatic speech recognizer. For this purpose, the open source speech-to-text recognizer *DeepSpeech* is used, which uses an acoustic model to translate the audio data into English text. In the second step, the output English text from DeepSpeech is fed into a character based n-gram model to classify the language as either of the four - Tamil, Odia, Assamese or Punjabi. Once these two processes are applied on the training data, the next step is to test the system. For this, audio from YouTube is used as test/evaluation data. Similar to the training data, the test audio data is also converted to English text. It is then given to the n-gram model, which classifies it into one of the 4 languages.

2 Related Work

In recent times, several methods have been developed to identify languages from audio/video data. Some of these methods use a speech recognizer to first convert the audio to text, and then train a model on the text to identify its language. Some methods perform a direct language identification from the raw audio data. For example, (Boussard et al., 2017) uses GMM and neural network models to effectively capture crucial information in the audio i.e, patterns they form and the sequence in which they are produced, rather than the sound waves. However this method is computationally expensive and requires more research on audio signal processing to include important features into the model. On the other hand, in this language identification system, since we convert audio to text before processing it, computation cost is very less and the overhead of understanding the audio signals is saved.

For conversion of audio to text, we use DeepSpeech, an open-source voice recognition that uses a neural network to convert speech spectrogram into a text transcript. Many recent works such as (Firmansyah et al., 2020) includes research work on DeepSpeech. Works like these show that DeepSpeech is suitable for less complex translation tasks.

For identification of language, we use an n-gram model in this paper. (Pohl and Ziółko, 2013) points out some disadvantages of such models and proposes a part of speech n-gram method for improving automatic speech recognition. This reference paper explains that the primary problem connected with n-gram based LMs is data sparsity – it is impossible to collect a corpus that would allow to compute the probabilities for any word sequence of a given length that might appear in the recognized speech. As a result there are sequences that lack probability estimation. Due to the fact that the number of tokens in inflectional languages is larger than in positional languages, this problem is amplified. Partial solution to this problem is usage of techniques such as smoothing, interpolation and backoff.

3 Dataset

The training data used for this work has been taken from Mozilla's Common Voice audio dataset (`https://commonvoice.mozilla.org/en/datasets`). Data has been procured for four uncommon Indian languages - Tamil, Odia, Assamese and Punjabi. The Tamil dataset consists of a total audio time of 20 hours, split into smaller audios of few seconds, spoken in 190 different voices. The Odia dataset consists of a total audio time of 5 hours, split into smaller audios of few seconds, spoken in 30 voices. Similarly, the Assamese dataset spans to a total of 0.66 hours, spoken in 11 voices and the Punjabi dataset comprises of 0.22 hours worth of audio, spoken in 7 voices.

The testing data for this system is taken from YouTube. Audios in the four languages Tamil, Odia, Assamese and Punjabi is downloaded using the python youtube_dl script.

4 Experiments

This section describes in detail, the end to end working of the language identifier. Figure 1 shows the architecture diagram of this language identification system.

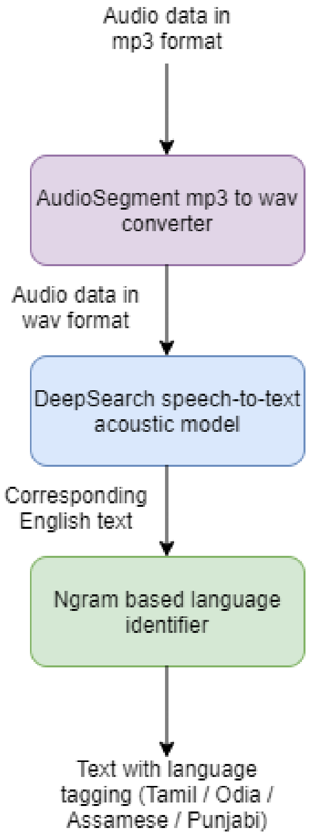
Audio data in mp3 format
AudioSegment mp3 to wav converter
Audio data in wav format
DeepSearch speech-to-text acoustic model
Corresponding English text
Ngram based language identifier
Text with language tagging (Tamil / Odia / Assamese / Punjabi)

Figure 1: Architecture Diagram

4.1 Data Preparation

The audio data from Common Voice, which is used as training data, is available in mp3 format. Since the automatic speech recognizer DeepSpeech uses wav audio format for speech to text translation, the mp3 audio files are converted to wav format using the python script *AudioSegment*.

4.2 Speech-to-text Conversion Using DeepSpeech

The audio data in wav format is then trained using DeepSpeech's pre-trained English model. This acoustic model will convert the audio data into English text. For training, 330 lines of Assamese, 3961 lines of Odiya, 135 lines of Punjabi and 12073 lines of Tamil have been used.

4.3 Language Identification Using Ngram Model

The converted English text is then fed into a character based n-gram language model. The model is first trained using the train data, and then it predicts the language on the test data. When the test data is given to the language model, it calculates the probabilities of the text being either of the four languages Tamil, Odia, Assamese or Punjabi and tags the text to the language with the highest probability (the language which matches closest to the text).

This approach uses the average normalised n-gram distances for most frequent bigrams, trigrams and quadgrams as a measure of similarity between the training data and the testing data. If there is a n-gram match with a language, the absolute value of the difference between its log probabilities is taken. All such absolute values are then summed up to get the distance of the given data from a language. In case a n-gram match is not found, the absolute value of the n-gram's log probability is used as a penalty term to penalize the result.

To detect the language of a document, at first its N-gram language model is created. Preprocessing is employed, i.e. punctuation marks are deleted and words are converted to lower case. Moreover, they are tokenized and surrounded with spaces or underscores. From these tokens, N-grams are generated and their occurrences are counted. The list of N-grams is sorted in descending order of their frequencies and the most frequent ones produce the N-gram language model of the document. (Panich, 2015) compares various language identification methods. The n-gram model implemented in our paper is based on this reference paper.

5 Results

For evaluating this system, test audio data is converted to English text and supplied to the ngram model. The test data comprises of 128 lines of Assamese, 95 lines of Odiya, 38 lines of Punjabi and 287 lines of Tamil. Random lines from random languages are selected to test the performance of the model.

Sentence	Actual	Predicted
a cal blo cup o pato a to eat blo ar	Odiya	Punjabi
hor wer on not non hoor an more terepor it	Tamil	Assamese
ah andwee rop yr a for lottle on herm of tro	Punjabi	Assamese
wo omer infaconye to to	Assamese	Punjabi
ruba wil wol a a a lo ho e	Tamil	Punjabi

Table 1: Misclassifications

Language	Accuracy	Correct	Total
Assamese	30.18%	16	53
Odiya	77.02%	57	74
Punjabi	90.00%	18	20
Tamil	54.74%	75	137

Table 2: Individual Language Accuracy

Accuracy is used as the metric to evaluate this language identification model.

$$Accuracy = \frac{NumberOfCorrectPredictions}{TotalNumberOfPredictions}$$

The highest accuracy achieved by the model is 50.67%, with 374 correct predictions out of 738 lines of data. Table 1 shows few examples of misclassifications.

Apart from randomly selecting lines from random languages, accuracy for each language is also calculated. Table 2 presents the results of this observation. It can be seen that Assamese has the lowest accuracy of 30.18% and Punjabi has the highest accuracy of 90%.

From the misclassifications observed, Assamese is recognized as Punjabi, Odiya is misclassified mostly as Punjabi and a few times as Tamil/Assamese. Punjabi is misclassified as Assamese and Tamil is misclassified as Punjabi/Assamese.

6 Conclusion

In this work, an n-gram model has been used to identify uncommon Indian languages - Tamil, Punjabi, Odia and Assamese. The audio is first converted to its English text using DeepSpeech's pre-trained English speech recognition model. This English text is then fed into a character based n-gram langugae model. This system achieves an overall accuracy of 50.67% on test audio data. Experiments were also done to see how the system performs on the four individual languages. The best recognized language was found to be Punjabi, with an accuracy of 90% and the worst recognized language was Assamese, with an accuracy of 30.18%. As a future work, further optimizations can be done to the n-gram model to improve accuracy. Also, instead of having an intermediate audio to text conversion module, features extracted from the raw audio can be supplied to a model to directly identify the language. This can improve the accuracy of the system and can overcome the errors produced by the speech recognizer.

References

Julien Boussard, Andrew Deveau, and Justin Pyron. 2017. Methods for spoken language identification.

Muhammad Hafidh Firmansyah, Anand Paul, Deblina Bhattacharya, and Gul Malik Urfa. 2020. Ai based embedded speech to text using deepspeech. *arXiv preprint arXiv:2002.12830*.

Leonid Panich. 2015. Comparison of language identification techniques. *Bachelor's Thesis, Heinrich Heine Universität Düsseldorf*.

Aleksander Pohl and Bartosz Ziółko. 2013. Using part of speech n-grams for improving automatic speech recognition of polish. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 492–504. Springer.