Interspeech 2018
2-6 September 2018, Hyderabad

An Automatic Speech Transcription System for Manipuri Language

Tanvina Patel, Krishna DN, Noor Fathima, Nisar Shah, Mahima C, Deepak Kumar, Anuroop Iyengar

Cogknit Semantics, Bangalore, India

{tanvina, krishna, noorfathima, nisar, mahima, deepak, anuroop}@cogknit.com

Abstract

Development of speech technologies in Indian languages has witnessed a steep improvement recently. In this work, we present our efforts in building various speech technology applications for Manipuri language. For the language at hand, we initially perform Language identification (LID) task. This is followed by speech-to-text (STT) and Keyword Search (KWS). In addition, we build a Speaker Diarization (SD) framework as well. The speech modules are integrated together to extract information from the speech signal. Currently, the platform is build for Manipuri and English language and can be extended to other languages as well. A User Interface (UI) is available for demonstration purpose where given a set of speech files the services from all the mentioned speech modules can be used.

Index Terms: Manipuri, LID, ASR, KWS and SD

1. Introduction

Development of speech technologies in multilingual societies such as India (with 22 major languages and over 1600 languages/dialects) is a challenging task. Over the years, efforts have been made to develop resources/data in Indian languages [1]. With the growing use of Internet and with the idea of digitalization, speech technology in Indian languages will play a crucial role in sectors like health care, agriculture, etc. [2]. On the similar lines, we present our efforts in developing a speech solution for Manipuri language. Manipuri (also known as Meitei) is an official language spoken in the northeastern part of India. It is a low-resource language, belonging to the Sino-Tibetan language family and is spoken by the inhabitants of Manipur and by the people at the Indo-Myanmar border [3].

In this paper, we present our efforts to develop a combined system comprising of speech technology applications like, Language Identification (LID), Speech-to-Text (STT), Keyword Search (KWS) and Speaker Diarization (SD). The proposed pipeline receives a set of audio files (possibly different languages). At first, the language of the audio files is identified using the LID module. Once the language is identified, the speech files are fed to the corresponding STT module to obtain the transcript, followed by KWS. Simultaneously, the audio files are passed through the SD framework to get diarized speech segments. These systems are useful in call centers were a lot of speech data is received in many languages and hence, it is essential to know the language and then direct it to an intended operator. The operator can also search through the KWs rather than listening all the files. This is one such application for the complete pipeline, however, many more can be related. The performance of ASR, LID and KWS, and SD modules are evaluated using Word Error Rate (WER), Equal Error Rate (EER), and accuracy, respectively. Details about the infrastructure used is given in [4]. Different tools and techniques are used to complete the pipeline and a visual interface is developed.

2. Developing the Transcription System

2.1. Speech Data Collection and Annotation

A data collection portal is build and speech corresponding to $\sim$100 hours is collected. The speech is telephonic in nature (sampled at 8000Hz) and recorded from 300+ native Manipuri speakers. The speech is read in nature and each speaker receives unseen 100-150 utterances corresponding to around 30 minutes.

Next, the speech is transcribed to check for any mismatch between the text and audio. Any empty files due to poor recording, etc. are removed. The speech is also tagged for non-speech parts and other fillers. The transcription was carried out by 5 trained Manipuri linguists in a period of around 6 months using wave-surfer as a tool. Once the text is annotated the lexicon is obtained through a rule based parser developed as part of TBT-Toolkit [5] that uses the Common Label Set (CLS) for phoneme representation across different Indian languages [6].

2.2. Language Identification (LID)

The LID module is an open-set system that classifies the test utterance to one of the four languages, i.e., Assamese, Bengali, Manipuri, English and an unknown class[1]. The KALDI toolkit and its Language Recognition Evaluation (LRE)'07 recipe is used [7]. That is, an i-vector based approach using full Gaussian Mixture Model-Universal Background Model (GMM-UBM) and logistic regression [8]. For training the LID module, 20 hours of speech data is used from each language, i.e., 100 hours comprising of $\sim$52000 utterances in total. The testing is carried on 500 utterances of each language for which the EER obtained is 5.88% and the Average Cost (Cavg) is 0.084.

2.3. Speech-to-Text (STT)

We build a Large Vocabulary Continuous Speech Recognition (LVCSR) system for Manipuri. The KALDI toolkit with the LibriSpeech recipe that uses speaker adaptive training is used. The CMU-Language Model (LM) toolkit is used to build a 2-gram LM [9]. The LM was build on $\sim$30k sentences, with an average of 10-15 words per sentence. Both Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) and Deep Neural Network-HMM (DNN-HMM) Acoustic Models (AMs) systems were built. The total cost in generating hypothesis is based on AM and the weighted LM cost. Further details about the features extracted and the parameters used are provided in [4],[10].

At present, the training data of 50 hours is used consisting of $\sim$65,000 words and $\sim$36000 sentences. The test data is of $\sim$11 hours and corresponds to $\sim$6500 words from about 60 speakers. The GMM-HMM system gives 19.28% WER and the DNN-HMM system gives better performance of 13.57% WER. For building the English STT system we use the 960 hours Librispeech corpus and its corresponding recipe as is.

[1]The Assamese and Bengali language data is obtained from the IARPA Babel Packs and for English language the Voxforge data is used.

2.4. Keyword Search (KWS)

Once the ASR decodes the speech, the KWS module, indexes the lattices and given a keyword/phrase, searches through the indexed lattices to get the occurrences of the desired KWs [11]. KWS gives better performance than ASR as it works on n-best lattices than the 1st best lattice. The KW set includes 100 unique unigram words randomly selected from the test set. There are a total of 4068 instances of the KWs in the test set. As compared to the 4068 instances of KWs in the test set, the detected KWs were 7688 and 7548 for GMM-HMM and DNN-HMM systems with 11.58% and 7.64% EER, respectively.

2.5. Speaker Diarization (SD)

The diarization module partitions an input audio according to the change in speakers. The LIUM diarization toolkit is used where for a given test speech, MFCC features are extracted and speaker segmentation is performed by first detecting instantaneous change points using Generalized Likelihood Ratio (GLR) distance. The distances between speakers is used to fuse consecutive segments that correspond to the same speaker. Hierarchical Agglomerative Clustering merges the two closest clusters at each iteration until the best Bayesian Information Criterion (BIC) distance is positive, followed by Viterbi decoding to generate a new segmentation using GMMs as speaker models [12].

3. Demonstration System

Figure 1 shows the User Interface (UI) developed to demonstrate LID, ASR, KWS and SD speech services. The top panel shows a view, where at the left, a set of wave-files appear with the language labels after LID. A word cloud is generated based on the KW list. The size of the text indicates the KW's frequency. The UI has options to click and play the selected wave-file. As the audio is played, the generated ASR transcript is displayed. If the text in the transcript matches with the KW list, then the KW is highlighted. It may happen that the ASR output may be erroneous and not match with the KW. In such cases (as shown in Figure 1 bottom panel), the KWS algorithm may identify a probable hit. The KWs are detected with a start and end duration, hence, in the UI, a feature is provided to click the transcript and the corresponding location in the audio panel is played. As shown in Figure 1, speaker changes are detected using SD and each speaker is highlighted. The platform is built in such a way that it can continuously take streaming data and return output in the form for language labels, transcription, KWs, etc. At the back-end we use big data technologies like, Kafka, HBASE, SOLR, etc. Everything is served in the form of APIs through NodeJs and the front-end UI is built with AngularJs.

4. Summary and Conclusions

In this paper, we demonstrate our efforts in developing various speech technology modules for Manipuri language and integrating them. The overall pipeline can be used as per the target user's needs and can also be replicated for other languages. In future we intend to integrate diarization with Speaker Identification (SID) module that will assist to identify if the speaker is one of the speakers from an existing database. We also have a Machine Translation (MT), i.e., native to English language support, however, it is beyond the current scope of submission.

[1]The authors thank Dinesh Wangkhem for managing the Manipuri recording/transcription task and Rajashree Jayabalan and the product engineering team at Cogknit for their efforts towards UI development.

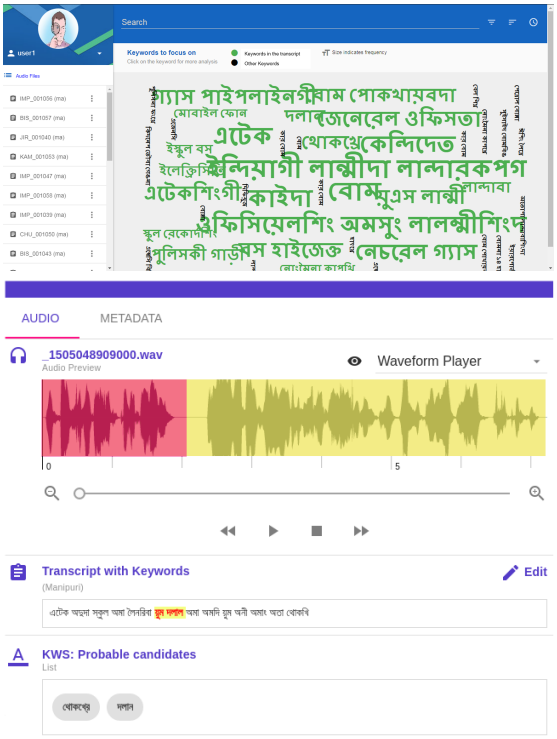

Figure 1: *Demonstration of LID, ASR, KWS and SD system.*

5. References

[1] G. A. Numanchipalli *et al.*, "Development of Indian language speech databases for large vocabulary speech recognition systems," in *SPECOM*, Patras, Greece, 2005, pp. 1–5.

[2] A. Mohan *et al.*, "Acoustic modelling for speech recognition in Indian languages in an agricultural commodities task domain," *Speech Communication*, no. 56, pp. 167–180, Jan 2014.

[3] Wikipedia. (2018) Meitei Language. [Online]. Available: https://en.wikipedia.org/wiki/Meitei_language

[4] D. N. Krishna *et al.*, "Automatic speech recognition for low-resource Manipuri language," in *GPU Technology Conference (GTC)*, SanJose, California, 26-29 March 2018.

[5] A. S. Ghone *et al.*, "TBT (toolkit to build TTS): A high performance framework to build multiple language HTS voice," in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3427–3428.

[6] IndicTTS. (2016) Indian Language Speech sound Label set (ILSL12). [Online]. Available: https://www.iitm.ac.in/donlab/tts/downloads/cls/cls_v2.1.6.pdf

[7] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *ASRU*, Hawaii, US, 2011, pp. 1–4.

[8] D. Martinez *et al.*, "Language recognition in ivectors space," in *INTERSPEECH*, Florence, Italy, pp. 861–864.

[9] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," in *EUROSPEECH*, 1997, pp. 2707–2710.

[10] T. B. Patel *et al.*, "Development of large vocabulary speech recognition system with keyword search for Manipuri," accepted in *INTERSPEECH*, Hyderabad, Sept. 2018.

[11] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2338–2347, Nov 2011.

[12] M. Rouvier *et al.*, "An open-source state-of-the-art toolbox for broadcast news diarization," in *INTERSPEECH*, Lyon, France, pp. 1477–1481.