System Description for Tagalog, Vietnamese, Javanese and Tamil OPEN ASR Challenge 2021

Bong Keun Yoo1, Ngoc Thuy Huong Thai1, Wiwik Karlina1, Jayakrishnan Melur Madhathil1, Yao Yang Hong1, Tuan Anh Hoang1, Hanwu Sun1, Huy Dat Tran1, Trung Tuan Luong1, Kah Kuan Teh1

1Institute for Infocomm Research, A*STAR, Singapore
E-mail: {yobk, nththai, wiwikk, jayakrishnan, hongyy, hoangta, hwsun, hdtran, luong-tt, tehkk}@i2r.a-star.edu.sg

Abstract
This system description describes the Automatic Speech Recognition (ASR) system for the Case Insensitive (CI) Open ASR challenge with Tagalog, Vietnamese, Javanese and Tamil Languages. We participate in constraint field of these languages. The data provided was 10-hours of ground-truth data for training and 10-hours development and 5-hours of evaluation data [1]. In addition to the provided data by Open ASR 2021, we also use Oscar text data [2] to interpolate with provided OPEN ASR 2021 text for building language models. To evaluate the performance of state-of-the-art speech and language systems for task oriented teams with naturalistic audio in challenging environments, we used data augmentation of speed perturbation on training dataset of OPEN ASR 2021. Various acoustic models (AM) were evaluated in conjunction with the n-gram based language models (LM) and Recurrent Neural Network (RNN) model.

Index Terms: automatic speech recognition, speech activity detection, OPEN ASR 2021

1. DATA RESOURCES
OPEN ASR 2021 distributes 10-hours of ground-truth data for training and 10-hours development and 5-hours of evaluation data for each language Tagalog, Vietnamese, Javanese and Tamil that we participated. In addition to the provided training dataset by OPEN ASR 2021, we also use OSCAR text data to interpolate with provided OPEN ASR 2021 text for building language models.

2. DETAILED DESCRIPTION OF ALGORITHM
In our proposed ASR system architecture for OPEN ASR 2021, we adopted data augmentation of speed perturbation for training an AM. To improve the accuracy of ASR system, we interpolate OPEN ASR 2021 training text data with the open source OSCAR text data to build better language model. After we built AMs and LMs, we obtained the final results through rescoring and lattice combination on factorized Time Delay Neural Networks (TDNN-F), Convolutional Neural Network (CNN)-TDNN-F, Time Delay Neural Network (TDNN)-Long Short Term Memory (LSTM) and CNN-TDNN-LSTM network architectures.

2.1. System overview
This ASR system was built using the open source Kaldi speech recognition toolkit [3]. The system was built on top of Linear Discriminant Analysis (LDA) [4], Maximum Likelihood Linear Transform (MLLT) [5], and feature space Maximum Likelihood Linear Regression (fMLLR) [6] features obtained from Gaussian Mixture Model (GMM) [7]. The 13-dimensional Mel-frequency Cepstral Coefficient (MFCC) features were extracted from audio and dimensionality reduction to 40 using LDA. On each frame, 100-dimensional i-Vector [8] was appended to the 40-dimensional LDA + MLLT + fMLLR with Cepstral Mean and Variance Normalisation (CMVN).

The network configuration for AMs were TDNN-f, CNN-TDNN-f, TDNN-LSTM and CNN-TDNN-LSTM architectures. The LMs were trained using 3-gram based LM and TDNN-LSTM with text data of training dataset and the open source OSCAR text data.

<table>
<thead>
<tr>
<th>Table 1: OSCAR text data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
</tr>
<tr>
<td>Tagalog</td>
</tr>
<tr>
<td>Vietnamese</td>
</tr>
<tr>
<td>Javanese</td>
</tr>
<tr>
<td>Tamil</td>
</tr>
</tbody>
</table>

2.2. Audio perturbation
Data augmentation is a common strategy adopted to increase the quantity of training data, avoid overfitting and improve robustness of the models. Speed perturbation [9] produces a warped time signal. Given an audio signal x(t), time warping by a factor α gives the signal x(αt). It can be seen from the Fourier transform of
For our participated four languages, Javanese, Tagalog, Tamil and Vietnamese, there are two type datasets: One is conversational telephone speech (CTS) and another is special distant mic recorded speech dataset. We applied two type VADs for their segmentation:

- From CTS data (with file extension .sph), we use a simple energy based VAD [11] to detect the starting and ending points and chunk the long voice data into short segments.
- For some special distant mic speech data with file extension .wav, we adopted the Sohn's statistical model based VAD to chunk the long wave files into segmentations [12].

2.3. VAD and segmentation

For training an LM we also use OSCAR text data to interpolate with provided OPEN ASR 2021 text for building language models.

2.4. Language Model

In addition to the provided training dataset by OPEN ASR 2021, we also use OSCAR text data to interpolate with provided OPEN ASR 2021 text for building language models.

2.5. Acoustic model

We used the same features as the baseline ASR system. These features were used inputs to the various network architectures including TDNN-F, CNN-TDNN-F, TDNN-LSTM and CNN-TDNN-LSTM. We evaluated the various network architectures on development set and evaluation set.

2.6. Results

Table 2 shows our WER results on dev and eval set of the four languages Tagalog, Vietnamese, Javanese and Tamil on constraint field and Case Insensitive.

Table 2: WER results

<table>
<thead>
<tr>
<th>Language</th>
<th>dev</th>
<th>eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagalog</td>
<td>0.5509</td>
<td>0.8365</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>0.5022</td>
<td>0.7980</td>
</tr>
<tr>
<td>Javanese</td>
<td>0.6273</td>
<td>0.8957</td>
</tr>
<tr>
<td>Tamil</td>
<td>0.8565</td>
<td>1.0170</td>
</tr>
</tbody>
</table>

3. HARDWARE REQUIREMENTS

The infrastructure used to run the experiments was 8 GPUs, Tesla V100-SXM2, 32GB each; and 40 CPUs, Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz. We used Kaldi toolkit with different deep neural network architectures for training AMs and decoding. System execution times to decode 60–70 minutes file vary depended on network architectures.

For training a model on CPU / GPU:
- Getting an alignment: 5–6 hours
- i-vector/ speed perturbation / getting new an alignment and tree: 6 hours
- Train an AM on DNN (about 3–5 hours for each network architectures (CNN-TDNNf, TDNN-f, TDNN-LSTM, CNN-TDNN-LSTM)

For training an LM
- n-gram based LM: less than 30 minutes
- TDNN-LSTM: less than 4 hours for training LM with training set and more than 72 hours for training LM with Oscar data on 8 GPUs

4. REFERENCES