Speech recognition on Indonesian language by using time delay neural network

Bagus Tris Atmaja[1,3*] , Fandy Akhmad N. F.[2], Dhany Arifianto[3], Masato Akagi[1]
[1]Japan Advanced Institute of Science and Technology, Japan
[2]Bahasa Kita Co., Ltd., Indonesia
[3]Institut Teknologi Sepuluh Nopember, Indonesia

1 Introduction

Speech recognition nowadays went from solved to unsolved again. It can be considered as solved for well-resourced language such as English, Spanish, Mandarin and Japanese, but still unsolved for (very) under-resourced language such as Malay, Indonesian, Iban and Javanese. Indonesian language, known as Bahasa, is one of under-resourced language due to limitation on its research and development. Although categorized as under-resourced language, the Indonesian language, combined with Malaysian language, is the 7th most spoken language by total numbers of speakers.

One of the most critical issues in speech and language research is speech recognition, i.e. how machine recognizes human speech. If the machine (computer) can perfectly understand what human says, it can be very useful for many other applications such as voice command, voice assistant, smart home system and robot audition. This is why we conduct this research. If a speech recognition system results high performance evaluation, it will improve the performance on other applications.

The speech recognition technique itself is rapidly developed from Hidden Markov Model (HMM) to Deep Neural Network (DNN). HMM modeled phones as Markov process to form state and observation. Given acoustic model, pronunciation dictionary and language model, HMM can decodes speech into words [1]. HMM is the most used speech recognizer system until the late of 1990s as it gives the highest performance measured as word error rate (WER) at that time. Besides the performance, other issues which have to be tackled in this research are computation time and effect of number of speakers.

The deep neural network is extension of neural network with deeper layer, commonly it requires more than 5 layers. A configuration of time delay neural network presented in [3] was proposed for Indonesian speech corpus. In a TDNN architecture, the initial transforms are learned on narrow contexts and the deeper layers process the hidden activations from a wider temporal context [3]. Hence, the more layer, the wider temporal context as context width increase as the higher layer. The TDNN algorithm is already implemented in Kaldi speech recognition toolkit which is used in this research.

To build speech recognition system on the top of Kaldi system, two main inputs are needed: pronunciation dictionary and language model. Then, an acoustic model can be built within Kaldi toolkit. Another important paper for this research beside [3] is [2]. We borrow language model and the recipe from [2] for Indonesian speech corpus, while the TDNN system is implemented as in [3]. The organization of this paper then will be continued by dataset used in the experiment, experiment setup, result and discussion and the last section is conclusion and future works.

2 Dataset

The Indonesian speech corpus which is taken from [4] is used in this research. There are 1529 utterances spoken by six speakers on the dataset. All speakers spoke the same utterances each other. Each utterance only consists of one sentence enounced by professional announcer and recorded in music studio as it is originally intended for speech synthesis. From those total 9174 utterances, we only take 1529 utterances to avoid learning repetition on the TDNN system. The distribution of the used utterances for each speaker can be shown in Table 1.

The set of the 1529 utterances is formed from 1029 declarative and 500 question sentences which the

*Corresponding author: Bagus Tris Atmaja (bagus@jaist.ac.jp)

Table 1 Number of used utterances per speaker

Speaker	Gender	Used utterances
1	female	250
2	female	250
3	female	250
4	male	250
5	male	250
6	male	279
	Total	1529

sentences are taken from movie and drama. The total utterances then are divided into two: training parts and development parts, which is explained in the next section. The details of the dataset can be found in [4].

3 Experiments

3.1 Dataset distribution

The obtained dataset first is processed into several parts. As stated previously, the other two inputs beside dataset is phone dictionary and language models. For phoneme dictionary, we only separate each words into its letters. For the language model, we borrow directly from Iban language model [2]. Hence, the presented result is not using Indonesian language as language model, but closely-related Iban language as language model. Those two inputs can be improved in future research.

The dataset obtained from [4] then is divided into training parts and development parts. For each speaker (saved in a directory) we took 200 utterances for training and 50 utterances for development except from the last speaker, in which we took 79 utterances for simplicity. Table 2 shows numbers of training and development utterances for each speaker.

The naming convention for the directory is ibXY_00Z where X is either f for female or m for male, Y is the initial (first letter of speaker) and Z is ordinal number of the speaker. Form that table, we can see the composition of train/development utterances is about 80/20.

3.2 Feature extraction

A number of 13 MFCC features are extracted from audio files consisting of 12 MFCC coefficients and energy. Deltas and deltas-deltas are also extracted resulting 39 features in total.

Table 2 Number of utterances for train and development

Directory	Train	Development
ibfa_001	200	50
ibfb_002	200	50
ibfe_003	200	50
ibme_004	200	50
ibmj_005	200	50
ibmm_006	200	79
Total	1200	329

For the TDNN technique, iVector is used to train the neural network, i.e normalization based on mean shift [3]. The iVector is obtained from,

$$M = m + Tw \tag{1}$$

where M is speaker and channel dependent supervector (feature extractor), m is speaker and channel supervector (obtained from a common Gaussian Mixture Model called Universal Background Modal) and T is rectangular low rank matrix, and w is the iVector (intermediate vector - total factors).

3.3 Training of Monophone, Triphone, and SGMM

On the experiment, we put monophone, triphone and subspace gaussian mixture model (SGMM) within a script and run it sequentially. Monophone is context-independent, all speech utterances are represented by concatenating a sequence of phone models together. This technique decomposes each vocabulary word into a sequence of context-independent base phones. Hence, it is difficult to capture the very large degree of context-dependent variation that exists in real speech [1]. Table 4 shows monophone approach has the highest WER among other approaches.

Triphone is context-dependent to solve the problem of monophone. This approach includes the neighbor phones by formula $x - q + y$ where x is the previous phones, q is the current phone and y is the next phone. For example word "stop" will be mapped as "SIL SIL-S+T S-T+OH OH-P+SIL". For this triphone, we used four variations: tri1a,

tri2a (tri1a+deltas+deltas), tri2b (tri2a + linear discriminant analysis (LDA)+ maximum likelihood linear transformation (MLLT)), and tri3b (tri2b + speaker adaptive training (SAT) + feature-space maximum likelihood linear regression (fMLLR)).

The last traditional approach is SGMM which is improvisation of HMM/GMM by deriving a common GMM called Universal Background Model (UBM) as acoustic unit. This approach takes the output of tri3b as input of UBM training. For the UBM training, the number of gaussian is set to 600. For the SGMM training as the last steps, the numbers of leaves and sub-states are 4200 and 6000.

3.4 Training of TDNN

Instead of feeding all inputs to neural networks in one go, TDNN feeds the inputs to the network as time series. The hyper-parameter in this architecture is input contexts of each layer required to compute an output activation, at one time step [3]. Fig. 1 shows the networks in TDNN while its hyper-parameters is defined in Table 3. For each layers, the dimension of 256 nodes are used equally. We use all hyper-parameters (sampling) instead of sub-sampled hyper-parameters as the size is small. While the number of used layers is 6, Fig. 1 shows 5 layers only as the 6th layer is the output layer with 0 hyper-parameter but it still has 256 nodes.

The activation function for all TDNN layers is ReLU renormalization layer. As we used only small data and single GPU, we set a small number for both initial and final jobs i.e. 2 and the number of training epoch is 3. For whole computation process, a number of 6 jobs is used to parallelize the computation as our CPU provided 10 cores. Our computing system used for this computation is i9-7900X CPU with GTX 1060 GPU with 64 GB of RAM under Ubuntu 18.04 operating system.

4 Result and Discussion

4.1 Performance of Indonesian speech corpus ASR

Kaldi toolkit [5] is used to evaluate TDNN for Indonesian speech corpus which steps are explained on the previous section. The result in %WER measure is shown in Table 4. This metric is obtained from the following formula,

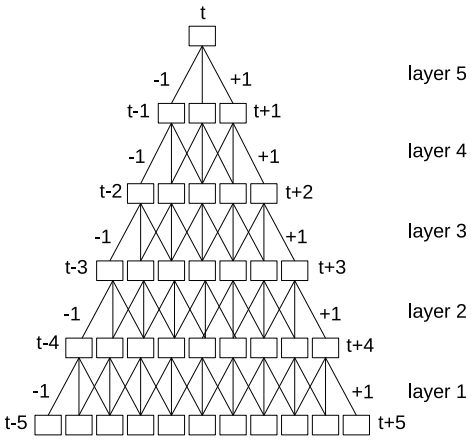

Fig. 1 Architecture of TDNN used to train Indonesian speech corpus.

Table 3 Hyper-parameters used on each TDNN layer

Layer	Input context
1	{-1, 0, 1}
2	{-1, 0, 1}
3	{-1, 0, 1}
4	{-1, 0, 1}
5	{-1, 0, 1}
6	{0}

$$WER = \frac{S + I + D}{N} \times 100\% \quad (2)$$

where S, I, D and N are number of substitution, deletion, insertion and total of words. The first three variables are recognized as error in word recognition.

It is clearly shown by the Table 4 that TDNN approach outperformed other traditional approaches. Beside the excellent algorithm inside TDNN, it also takes the output of SGMM so it has more information among other approaches. This result also supports the previous research [2] reported that the related language can be used to develop ASR for a language. While [2] used Malay language to develop Iban ASR, this research presented the development Indonesian ASR using Iban language model. Both Indonesian and Iban are rooted from Malay language.

4.2 Computation Time

One of motivations to use TDNN is to shorten the computation time. The required computation time

Table 4 Performance of Indonesian speech corpus ASR among different approaches

Training approach	%WER
Monophone	37.17
Triphone 1	26.53
Triphone 2a	26.06
Triphone 2b	26.33
Triphone 3b	22.29
SGMM	18.22
TDNN	17.54

Table 5 Effect of number of speakers

Number of speakers		Separation	%WER
Train	Dev		
4	2	Yes	33
6	6	No	17

for TDNN on our system is about 10 minutes, while total running time for all non-TDNN approaches is about 16 minutes (non-TDNN result must be obtained first to run TDNN). By using our dataset, we reach the similar WER score using only 26 minutes of TDNN training time compared to 7 hours of DNN training time shown in [2].

4.3 Effect of Number of Speakers

In addition of obtained ASR, we also spotted the effect of trained number of speakers. Before we obtained result as shown in Table 4, we got poor result of TDNN approach due to lack of number of speakers: four speakers for train and two speakers for development. Finally, we use all six speakers for both train and development but by using different utterances. Table 5 shows the different number speakers in train/dev and its WER. Separation column shows if the speakers used for train is also used for development ("No") or not ("Yes"). If more speakers are available on the future dataset, it should be separated for train and development portion to make the system robust against speaker differentiation. TDNN itself takes advantage of number of speakers as iVector feature capture both speaker and environment specific information [3].

We also conducted TDNN experiment with all 9174 utterances data which resulted very small WER, about 3%. This result cannot be guaranteed as real WER as one utterance is repeated by six speakers which makes algorithm train the same word in the same sentences six times.

5 Conclusion

We presented a simple yet promising approach to build Indonesian speech recognition by using TDNN under Kaldi toolkit. Although it used Iban as language model, the result shows potential improvement for the development of Indonesian ASR. The WER metric as evaluation method shows TDNN reaches the lowest score among other approaches.

Our computation time used to obtain the TDNN result shows inexpensive load compared to DNN system. It means there is no need to use supercomputer or high performance computer (HPC) cluster to reproduce our TDNN result. The recipe to reproduce this experiment is openly available in the repository[1] for result reproduction except for the dataset.

The last evaluation shows that the number of speakers play important roles in WER result. The more number of speakers, the the better WER obtained. We propose to add the number of speakers for the future works including expressive speech corpus.

References

[1] M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition," Found. Trends Signal Process., vol. 1, no. 3, pp. 195–304, 2007.

[2] S. S. Juan, L. Besacier, B. Lecouteux, and M. Dyab, "Using resources from a closely-related language to develop ASR for a very under-resourced language: A case study for iban," Proc. INTERSPEECH, pp. 1270—1274, 2015.

[3] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," Proc. INTERSPEECH, pp. 2440–2444, 2015.

[4] E. Cahyaningtyas, and D. Arifianto, "Development of Under-Resourced Bahasa Indonesia Speech Corpus," in APSIPA-ASC, 2017.

[5] D. Povey et al., "The Kaldi speech recognition toolkit," IEEE ASRU Workshop, pp. 1–4, 2011.

[1] https://github.com/bagustris/id