Towards Transfer Learning for end-to-end Sinhala Speech Recognition by Finetuning Pretrained Models

Anonymous ACL submission

Abstract

With the recent advancement, automatic speech recognition (ASR) moves toward addressing low-resource speech recognition problems using large vocabulary continuous speech recognition (LVCSR). Transfer learning, meta-learning, and Unsupervised Pre-training are major techniques in the modern paradigm, and in this paper, We experimented with transfer learning using the English pre-trained model trained on top of the Recurrent Neural Network (RNN) with the baseline e2e Lattice-Free Maximum Mutual Information (e2e LF-MMI) model models with 200 hours of OpenSLR data and 40 hours of gathered Sinhala speech data. We used Facebook Sinhala main corpora and UCSC full corpus alongside the UCSC speech corpus to train the external language models. We were able to achieve 5.43% WER for our testing dataset by far the best wer achieved for Low Resourced Sinhala Language. Finally, we evaluated the best e2e model with Google speech recognition API for Sinhala Speech Recognition using a publicly available dataset to examine how far we can use our model in common usage.

1 Introduction

End-to-end ASR systems are vastly improving over the traditional statistical models since 2019 where it gives the correspondent text in character or phone level directly from the given audio using a larger DNN without using an additional pronunciation model and language model. Since e2e ASR is created on top of a large deep neural network, it can be used to improve the recognition of low-resource language by transfer learning from high-resource languages like English. Traditional statistical systems like GMM-HMM have not supported transfer learning but DNN-HMM like statistical models can also be used but the e2e models show higher accuracy (Wang et al., 2019).

The Sinhala language is a low-resource language and it belongs to the Indo Aryan language family. Statistical ASR systems can be found for open domain Sinhala speech recognition and state-of-the-art results are yet to be achieved (Gamage et al., 2020; Karunathilaka et al., 2020).

In this paper, we are focusing on achieving higher results through doing Transfer learning from the English pre-trained model to Sinhala using RNN architecture created using Deepspeech. The performances of each e2e model will be evaluated and compared with e2e LF-MMI architecture using Sinhala Speech Recognition models.

The paper is organized as follows. Section 2 presents the related studies, Section 3 describes the methodology, data preparation, and implementation in greater detail. Section 4 describes the results and evaluation. Section 5 presents the conclusions and future work.

2 Related Work

The current domain of speech recognition moves toward addressing the low-resource problem. There are large datasets available for English and France like languages with state-of-the-art results. A common solution for addressing the low-resource problem is to transfer learning from high resource language to a low-resource language and even Deepspeech also provided scripts to transfer learning using common voice data for English language which has 2181 hours of training data. In the e2e LF-MMI technique transfer learning can be done by using weight transfer and multi-task training (Ghahremani et al., 2017).

Also in 2020, much research has been conducted about improving low-resource speech recognition by doing transfer learning techniques. Analyzing ASR Pretraining for Low-Resource Speech-to-Text Translation, Transfer Learning for Less-Resourced Semitic Languages Speech Recognition: the Case of Amharic, LTL-UDE at Low-Resource Speech-to-Text Shared Task: Investigating Mozilla DeepSpeech in a low-resource setting (Stoian et al.,

2020; Woldemariam, 2020; Agarwal and Zesch, 2020) are some major research conducted in these domains and ASR models of these papers have used LF-MMI model and Deepspeech models for their research which are published in the year 2020.

3 Approach

3.1 Data Preparation

The Deepspeech toolkit is not based on phonemes and it gives character sequences from the wav files that we fed into the RNN. The pronunciation model can not be visible and RNN consists of both the acoustic model and the pronunciation model. The Sinhala language has a good pronunciation model presented in (Nadungodage et al., 2018) paper but we did not focus on creating a lexicon for the pronunciation model in this study.

3.1.1 Dataset

We have used two datasets for the experiment. Open SLR Sinhala ASR dataset was used for transfer learning and LTRL speech data was used for finetuning the already transfer learned models.

1. **Large Sinhala ASR training data set by Open SLR**

   Open SLR has gathered Sinhala ASR training data set containing ∼185K utterances. But there are some miss transcriptions, numerical values, and foreign language words, especially in English. In the preprocess we have removed the utterance containing numerical values and foreign language transcriptions. Overall there were ∼181 hours of preprocessed utterances (Kjartansson et al., 2018).

2. **LTRL Speech Data (LSD)**

   we have used the collected recordings from the Language Technology Research Laboratory (LTRL) of the University of Colombo School of Computing (UCSC) which has 40 hours of training data. Training has been done in 16kHz sample rate and refers (Gamage et al., 2020) for more details. This dataset tagged with the gender and details is in table 1. We used this dataset to finetune the transfer learned models as the transcriptions of the utterances are more trusted with respect to the OpenSLR. The test set specified here was gathered in a noisy environment using a mobile phone and we have used that data for evaluations.

Dataset	Male	Female	Utterances
Train	27	67	17848
Dev	3	8	2002
Test	4	4	80

Table 1: Details of train, validation and test data sets

3.1.2 Corpora

We have used 3 corpora to train the language models in the study namely LTRL speech corpus, Facebook Sinhala main corpus (Wijeratne and de Silva, 2020), and transcriptions of OpenSLR preprocessed dataset.

1. **LTRL Speech Corpus (LSC)** is created using an active learning method and baseline models are used 4-gram language model created by this corpus (Gamage et al., 2021).

Vocabulary Size	243339
Total number of Sentences	119621
Total number of words	1194940

Table 2: Corpus Statistics of LTRL speech corpus

2. **Facebook Sinhala main corpus (FBC)** is used for annotation in FastText to train the embedding and we selected this corpus to achieve more accuracy in the decoding.

Vocabulary Size	228533
Total number of Sentences	3642053
Total number of words	∼34 million

Table 3: Corpus Statistics of Facebook Sinhala main corpuss

3. **OpenSLR speech corpus** has the transcriptions of its own utterances. We have concatenated all the 3 corpora for a single corpus to train the scorer model which is the language model in Deepspeech that act as an external language model when decoding. here we have used 5-gram language model to train the scorer.

3.2 Baseline models

We have selected baseline models presented in (Gamage et al., 2021) which use e2e LF-MMI architecture to train the e2e model. 28.55% WER was the best WER achieved for Sinhala Speech Recognition based on the same testing dataset that we have used in this study also. Table 5 represents

Vocabulary Size	64258
Total number of Sentences	149856
Total number of words	659086

Table 4: Corpus Statistics of OpenSLR speech corpuss

the WERs achieved in e2e LF-MMI models. these models are trained on the Kaldi toolkit and use LTRL speech corpus in creating language models.

Ephocs	Test set (WERs)
10	28.55
30	32.18
50	33.27

Table 5: WER comparison of baseline e2e LF-MMI models

3.3 RNN Architecture

RNN in Deepspeech does not use the so-called phones to train models (Hannun et al., 2014). Instead as mentioned earlier, it uses an alphabet of the training language to get the character sequence through a large DNN. And also separate n-gram language model can be used to decode utterances and in Deepspeech documentation[1] it is mentioned as External Scorer.

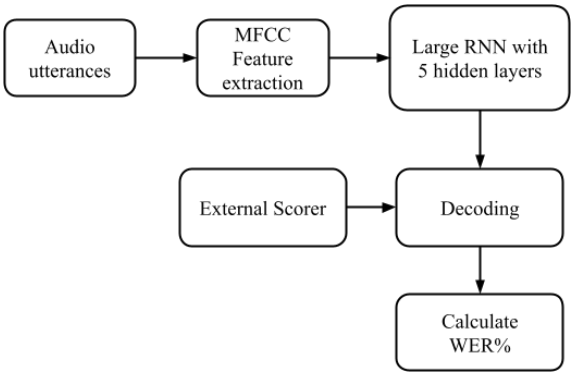

Figure 1: Deepspeech RNN model Architecture

26 MFCC feature extraction is done here, which is the standard setting for the 16kHZ sample rate used in Deepspeech (Hannun et al., 2014). Extracted features are fed to the first 3 non-recurrent layers which use Rectified-Linear (Relu) activation function. The Fourth layer is a recurrent layer including hidden units with forward recurrence. The fifth layer also a non-recurrent layer takes forward units as inputs. The output layer will predict the character probabilities for each time slice. The created external scorer can be used to have a more accurate output and then we calculated the WER with the testing dataset. The basic structure is represented in figure 1 and we created models for 30, 50, and 100 epochs.

After training, an output_graph.pb model file is generated which has extra loading time and memory consumption because the model needs to be loaded in memory to be dealt with when running inference. One way to avoid this is to directly read data from the disk. TensorFlow has tooling to achieve this.

We used the default 6 layers NN architecture where the recurrent occurs in the 4th layer to train the RNN models. 375 hidden units are in each hidden layer and the default RNN architecture in Deepspeech represented in (Hannun et al., 2014) is used through the study with the alphabet taken from the Sinhala Unicode character table. Zero-width space, zero-width joiner, and zero-width non-joiner characters had to be included in the alphabet to mitigate the errors occurring when training for the Sinhala language.

Finally, we used an external scorer to get higher accurate decoding. We have used the previously mentioned 3 corpora to train the scorer model and all models are finetuned to 1000 iterations. The scorer model can be finetuned by using the two parameters Lm_alpha, and LM_beta where Lm_alpha is for determining how much the language model is allowed to edit the network output and LM_beta controls inserting spaces (Hannun et al., 2014).

3.4 Transfer learning

With the limitations in dataset, transfer learning has been successful when creating an ASR system (Kunze et al., 2017). We have used the v0.9.3 English pretrained model[2] in this study as the source model and we replaced the output layer which consisted of English alphabet to Sinhala alphabet as the source output layer is not important (Ardila et al., 2019). We used the OpenSLR speech data to transfer learn the English pre-train model to the Sinhala Language. OpenSLR covers the basic phones where it has found text prompts online.

3.5 Finetuning pretrained models

After the transfer learning, we used LTRL speech data to finetune the RNN. LTRL speech data was

[1] https://deepspeech.readthedocs.io/en/v0.9.3/

[2] https://github.com/mozilla/DeepSpeech/releases/tag/v0.9.3

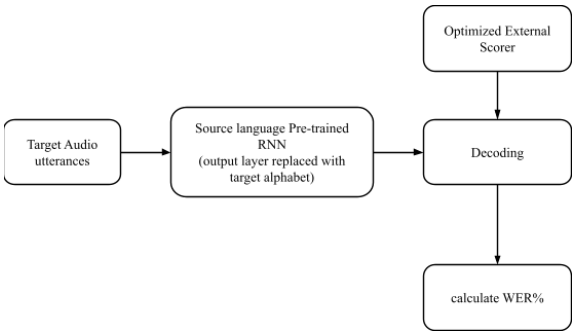

Figure 2: Transferlerning with OpenSLR dataset

gathered using an active learning method to cover all Sinhala phones. Data size is limited to 40 hours and it is not enough to get a state of the art results through direct e2e training (Wang et al., 2019). But (Gamage et al., 2021) paper shows that the data is more context-dependent in News and Number readings in the Sinhala Language. Here we have retrained the RNN which was previously trained on English and Sinhala OpenSLR data so we can have a better more general ASR system for Sinhala. The same Sinhala Alphabet which has been used in training the basic RNN for Sinhala has been used in Transferelearning and Finetuning processes.

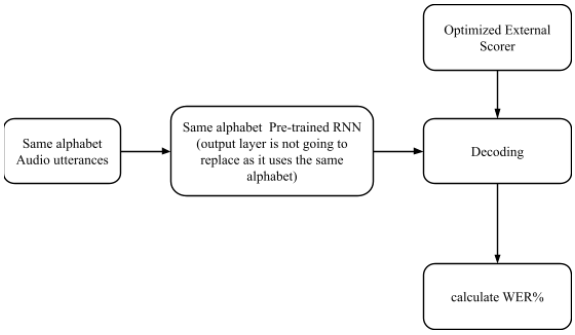

Figure 3: Finetuning Pretrained Models using LTRL speech data

4 Results and Evaluation

The accuracy of the models is measured using Word Error Rate (WER).

4.1 RNN models

The Deepspeech RNN model is a fully e2e model so it is not using HMM. The concept of phone is rejected here and uses a character-level alphabet instead. Table 6 represents the results that we achieved in the Deepspeech RNN Sinhala e2e ASR system. The literature highlighted that the e2e models need to have higher data with respect to good accuracy.

Model	epochs 30	epochs 50	epochs 100
RNN Model	54.83	43.80	47.43

Table 6: WERs of e2e RNN model

4.2 Transferlearned models

So we used the transfer learning technique provided by the Deepspeech toolkit for achieving lesser WER for Sinhala. We have used both LTRL speech data and OpenSLR dataset for transfer learning and the best results we obtained shows in table 7.

Model	Corpora	epochs	WERs
pretrain+ LSD	LSC	75	23.38
Pretrain+ OpenSLR	LSC+OpenSLR	10	17.04
	LSC+OpenSLR+ FBC	10	7.24

Table 7: WERs of transfer learned models using English Pretrained RNN

We can clearly identify that the larger scorer model is providing higher accuracy where we can get around 10% more accurate model using the Facebook Sinhala Corpus. We also trained the pretrain+openSLR model for 50 epochs but WER was 7.84% which is slightly higher than what we got for 10 epochs presented in table 7.

4.3 Finetuned Pretrained models

After we finetuned the models using LTRL speech data which were previously learned on top of English and OpenSLR Sinhala Dataset, we saw that there was a 2.12% increase in WER when we use the LTRL speech corpus concatenated with OpenSLR transcription. Results of finetuned models are shown in table 8.

Corpus	T_epochs and F_epochs	WERs
LSC+OpenSLR	10_75	19.16
LSC+OpenSLR +FBC	50_50	05.43

Table 8: WERs of Finetuned models using LTRL speech Data with the epochs of both transfer learned (T_epochs) models and finetuned(F_epochs) models

When we look into the worst decodings of all the transfer learned and pre-train models and we

saw that the transcriptions are accurate but we are getting higher WERs due to the technique in the Sinhala Language called 'Pada Bedima' which is addressed well in the evaluations of the paper. So actual WERs are lesser.

4.4 Evaluation

Google Speech Recognition API has an ASR system for the Sinhala language which is currently used for public-domain speech recognition tasks in Sinhala. We used two datasets to evaluate e2e LF-MMI model and our Finetuned RNN model with Google to find out how far our model can be used for general purposes. Those datasets are ones we used to test the created models mentioned above and we use a publicly available dataset that was used to create a Sinhala Text-To-Speech (TTS) system [3]. There were 3300 utterances available and we used the first 100 utterances after removing utterances gathered in the context of Pali and Sanskrit. Because those utterances have a different accent from Sinhala which may mislead the results from a general aspect.

Table 10 shows there are higher WERs for the Sinhala TTS dataset because that dataset was gathered for the context of creating the Sinhala Text-to-Speech system. So They have used a technique called 'Pada Bedima' (Rajapaksha, 2008) which is a collection of rules for segmenting Sinhala words. But Vocabulary of the Sinhala Language accepts the words without using the 'Pada Bedima' technique. More information is available (Liyanage et al., 2012) paper about the 'Pada Bedima' under the Discussion section. We did the analysis upon finding the percentage which cost the accuracy with the Sinhala TTS dataset.

As shown in Table 11 from the error words there are 32.08% words for 'Pada Bedima' in e2e LF-MMI model and 31.62% words for the fine-tuned model. We have removed them when computing WER so overall 25.42% WER from Finetuned model and Google achieved 26.94% WER for the Sinhala TTS dataset. E2e LF-MMI model has more accuracy in number readings especially phone numbers, credit cards, and time when observing the results but Our Finetuned model performed in a more general context and the given output are more accurate compared to google but especially google has more speed and able to recognize proper nouns better.

[3] https://github.com/pathnirvana/sinhala-tts-dataset

5 Limitations

Even though the results show 05.43% WER this still can be biased towards the testing dataset we have used. Based on our evaluation, the model performs better than the Google model for Sinhala by 1.52%, but it still cannot achieve the state-of-the-art word-error rate. This implicates the RNN needs more speech data in order to achieve higher results so the proposed model is still not the best respect to the available data. The server we used to train models had 4 RTX 2080TI GPUs. The training took 14 days to complete so we had a hard time finetuning each model. So we have used only the language parameters(alpha and beta) for the finetuning and further training can be done using the augmentations of was files as well. This will take a considerable time, and with that limitation, we did not consider them for this research.

6 Conclusion and Future Work

We were able to achieve the so far the best WER for Sinhala speech recognition through finetuning the pre-train model for a public domain which has 05.43% WER. This model performs more context independently whereas other public domain Sinhala speech recognizers have more context-dependent in areas like NEWS reading and Number readings (Gamage et al., 2020, 2021; Karunathilaka et al., 2020).

Further improvements can be achieved through multilingual speech recognition techniques (Conneau et al., 2020). As Sinhala belongs to Indo Aryan language family and using the dataset of those languages on top of multilingual low-resource self-supervised and unsupervised pre-training techniques can be used for future research on improving Sinhala Speech Recognition to achieve state-of-the-art results.

References

Aashish Agarwal and Torsten Zesch. 2020. Ltl-ude at low-resource speech-to-text shared task: Investigating mozilla deepspeech in a low-resource setting.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Model	ephocs	WER of Testing Data		
		LSC	LSC+OpenSLR	LSC+OpenSLR+FBC
LSD	50	41.93	53.99	51.28
Pretrain + LSD	75	23.38	13.53	26.84
Pretrain + openSLR	10	11.31	17.04	07.23
Pretrain + openSLR + Fine-tuned with LSD	50	11.46	19.60	05.43

Table 9: WER of all trained models over the corpus used for decoding

Model	Testing Dataset	Sinhala TTS Dataset
E2e LF-MMI	28.55	46.72
pre-train+openSLR+ LTRL speech Data	5.43	38.32
Google Speech API for Sinhala	36.07	39.73

Table 10: WERs of e2e LF-MMI model and RNN Fine-tuned model with Google Speech Recognition API for the Sinhala language.

Model	Percentage for 'Pada Bedima'	Overall WER
E2e LF-MMI	32.08	30.62
pretrain+openSLR+ LTRL speech Data	31.62	25.42
Google Speech API for Sinhala	29.40	26.94

Table 11: Percentage of words for 'Pada Bedima' over error words

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Buddhi Gamage, Randil Pushpananda, Thilini Nadungodage, and Ruvan Weerasinghe. 2021. Improving sinhala speech recognition through e2e lf-mmi model. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, National Institute of Technology Silchar, Assam, India. NLP Association of India (NLPAI).

Buddhi Gamage, Randil Pushpananda, Ruvan Weerasinghe, and Thilini Nadungodage. 2020. Usage of combinational acoustic models (dnn-hmm and sgmm) and identifying the impact of language models in sinhala speech recognition. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 17–22. IEEE.

Pegah Ghahremani, Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur. 2017. Investigation of transfer learning for asr using lf-mmi trained neural networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 279–286. IEEE.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hirunika Karunathilaka, Viraj Welgama, Thilini Nadungodage, and Ruvan Weerasinghe. 2020. Low-resource sinhala speech recognition using deep learning. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 196–201. IEEE.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, pages 52–55, Gurugram, India.

Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017.

Transfer learning for speech recognition on a budget. *arXiv preprint arXiv:1706.00290*.

Chamila Liyanage, Randil Pushpananda, Dulip Lakmal Herath, and Ruvan Weerasinghe. 2012. A computational grammar of sinhala. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 188–200. Springer.

Thilini Nadungodage, Chamila Liyanage, Amathri Prerera, Randil Pushpananda, and Ruvan Weerasinghe. 2018. Sinhala g2p conversion for speech processing. In *SLTU*, pages 112–116.

D Rajapaksha. 2008. Sinhala bhashave pada bedima saha virama lakshana bhavithaya. *Dharma Rajapaksha*.

Mihaela C Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing asr pretraining for low-resource speech-to-text translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913. IEEE.

Dong Wang, Xiaodong Wang, and Shaohe Lv. 2019. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018.

Yudhanjaya Wijeratne and Nisansa de Silva. 2020. Sinhala language corpora and stopwords from a decade of sri lankan facebook. *arXiv preprint arXiv:2007.07884*.

Yonas Woldemariam. 2020. Transfer learning for less-resourced semitic languages speech recognition: the case of amharic. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 61–69.