Deep learning based large vocabulary continuous speech recognition of an under-resourced language Bangladeshi Bangla

Ahnaf Mozib Samin*, M. Humayon Kobir†, Shafkat Kibria‡ and M. Shahidur Rahman§

Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet-3114, Bangladesh

(Received 13 November 2020, Accepted for publication 13 April 2021)

Abstract: Research in corpus-driven Automatic Speech Recognition (ASR) is advancing rapidly towards building a robust Large Vocabulary Continuous Speech Recognition (LVCSR) system. Under-resourced languages like Bangla require benchmarking large corpora for more research on LVCSR to tackle their limitations and avoid the biased results. In this paper, a publicly published large-scale Bangladeshi Bangla speech corpus is used to implement deep Convolutional Neural Network (CNN) based model and Recurrent Neural Network (RNN) based model with Connectionist Temporal Classification (CTC) loss function for Bangla LVCSR. In experimental evaluations, we find that CNN-based architecture yields superior results over the RNN-based approach. This study also emphasizes assessing the quality of an open-source large-scale Bangladeshi Bangla speech corpus and investigating the effect of the various high-order N-gram Language Models (LM) on a morphologically rich language Bangla. We achieve 36.12% word error rate (WER) using CNN-based acoustic model and 13.93% WER using beam search decoding with 5-gram LM. The findings demonstrate by far the state-of-the-art performance of any Bangla LVCSR system on a specific benchmarked large corpus.

Keywords: Large vocabulary continuous speech recognition, Convolutional neural network, Recurrent neural network, Language modeling, Bangladeshi Bangla

1. INTRODUCTION

Bangla is the native language of around 228 million people and around 37 million people use it as their second language. It is the 7th most widely spoken language in the world. But research on Bangla speech recognition is far behind than other languages such as English, Mandarin Chinese, Japanese, Korean and some European languages etc. [1,2]. There are mainly two types of accents in Bangla language: Bangladeshi standard Bangla and Kolkata (capital city of West Bengal, India) standard Bangla. In 2011, a speaker independent Kolkata standard Bangla speech corpus “Shruti” of 21.64 hours is developed by the Communication Empowerment Lab, Indian Institute of Technology, Kharagpur (IITKGP) [3]. They use Hidden Markov Model Toolkit (HTK) [4] to build mono-phone acoustic model to test phoneme recognition accuracy.

Moreover, they develop tri-phone model using CMU-SPHINX toolkit [5] to detect word recognition accuracy. Both of these toolkits are based on Gaussian Mixture Model-Hidden Markov Model (GMM-HMM). M. A. A. Amin et al. [6] build Deep Neural Network-Hidden Markov Model (DNN-HMM) and GMM-HMM based ASR systems on the same corpus using Kaldi toolkit [7]. They utilize improved feature extraction technique provided by Kaldi. However, they conclude that 21.64 hours of speech corpus is not sufficient for training a DNN-HMM based model. In 2016, 215 hours of Bangla telephone conversation based annotated speech corpus is developed under the IARPA (Intelligence Advanced Research Projects Activity) Babel program [8]. This corpus also contains Kolkata accented Bangla speech which is significantly different from Bangladeshi standard Bangla. In 2018, an RNN-based noise robust system is built by S. H. Sumit et al. [9]. They develop 300 hours of “Sociam” speech corpus which is not publicly available and add 50 hours from Babel Bangla corpus and then train an acoustic model on a total of 350 hours of dataset. As Sociam corpus is not a publicly published benchmarked corpus, it is not possible
A. M. SAMIN et al.: DEEP LEARNING BASED LVCSR OF BANGLADESHI BANGLA

for the research community to conduct experiments and measure the quality of the corpus. According to them, their WER on clean read speech is 53.69% without data augmentation and 32.19% after augmentation. In 2020, S. Ahmed et al. [10] present an approach for automatic preparation of speech corpus in Bangla language utilizing publicly published Google OpenSLR (Open Speech and Language Resources) Bangla corpus [11] and open-source audiobooks and TV news recordings. They prepare 960 hours of annotated speech corpus and train GMM-HMM model using Kaldi and Long Short-Term Memory (LSTM) network with hybrid CTC-attention mechanism. Like Socian corpus, the newly prepared corpus by [10] is not a benchmark one as the correctness of annotation is not confirmed by human supervision and also the corpus is not publicly available. There are other works done on small datasets in Bangla language but in this paper, our discussion is limited to only LVCSR systems.

The main challenges in developing a robust Bangla LVCSR system are accent variability in Bangla language [12,13], complexity of morphological parsing [14] and lack of large corpora. The only publicly available large annotated corpus in Bangladeshi Bangla is Google OpenSLR Bangla corpus which is approximately 210 hours long [11]. G. Hinton et al. [15] showed around 300 hours of speech data is required for DNN-HMM based models to outperform GMM-HMM models. So, the following questions arise: for an under-resourced language like Bangla, what can be done to build a robust ASR system considering the constraints? Can exploring various deep neural network architectures such as CNNs, variants of RNNs such as Gated Recurrent Unit (GRU) and RNN-Transducer, attention-based neural networks etc. mitigate the limitations of Bangla language? Or instead of focusing only on acoustic modeling, is more attention needed to language modeling as well to build a robust LVCSR system?

In this paper, we propose deep CNN with CTC loss function for continuous Bangla speech recognition. As far as we know, this is the first time fully CNN-based LVCSR system is exploited in Bangla language. Moreover, we also train a RNN-CTC based model. Our RNN-based model is the implementation of Deep Speech 2 (DS2) system first proposed by Baidu Research Lab [16]. In their paper, they implement this architecture for English and Chinese Mandarin language but we adapt it for Bangla LVCSR and compare it with the CNN-based approach. These two approaches have been utilized to evaluate currently the largest and only publicly published Bangladeshi Bangla speech dataset — Google OpenSLR Bangla speech corpus. In addition, we build a large-scale N-gram LM with 10.6 million vocabulary. To our knowledge, this is the largest LM integrated to an ASR system in this language. Our major contributions can be summarized as follows:

- We show that our proposed end-to-end deep CNN with CTC based Bangla ASR system outperforms RNN-based DS2 system on Google Bangla corpus (See Sect. 6.3).
- We experiment with several high-order large-scale N-gram LMs. Our 5-gram LM further improves the performance of our system by reducing WER from 36.12% to 13.93%. We demonstrate that hyperparameter tuning of the beam decoder is an important factor that helps us get the state-of-the-art result (See Sect. 6.4). These experiments not only improve Bangla ASR research but also motivate other under-resourced languages which are also lacking large annotated speech corpora.
- We assess the quality of the Google Bangla speech corpus by analysing character-wise error rates using deep learning approaches. To the best of our knowledge, this kind of analysis on a specific corpus is done for the first time in Bangla ASR research (See Sect. 6.5).

The outline of this paper is as follows. We discuss our input feature extraction techniques in Sect. 2. In Sect. 3, our CNN-based and DS2 architectures are described. Two approaches of decoding are mentioned in Sect. 4. Description of our speech and text corpus can be found in Sect. 5. In Sect. 6, we are going to explain the experiments and analyse the results. A conclusion and future scope is presented in Sect. 7.

2. FEATURE EXTRACTION

The first step is to extract acoustic features from the raw signal using a feature extraction method that eliminates noise and other irrelevant information and contains mostly linguistic contents. We use Mel Frequency Cepstral Coefficient (MFCC) features for our CNN-based model which is widely used acoustic features in speech recognition and speaker identification. For windowing the non-stationary speech signal into small frames, proper frame size and stride need to be determined so that better spectral feature extraction of sound can be performed. We experiment with different frame sizes and strides to see how it impacts WER (See Sect. 6.2). Then, power spectrum of each frame is calculated by applying Fast Fourier Transformation (FFT). After that, we apply 40 filterbanks to the power spectrum using mel-scale. Finally, after taking the logarithm of the power spectrum, Discrete Cosine Transform (DCT) is applied to calculate MFCC coefficients. We choose 21 MFCC coefficients for our best model. Then these MFCC feature vectors are used as input to the acoustic model.

To train our RNN-based DS2 model, spectrogram features are fed as input to the neural network.
3. ACOUSTIC MODELING

3.1. Convolutional Neural Network
We train an acoustic model using Convolutional Neural Network (Fig. 1). The number of convolutional layers is 20. Each convolutional layer has 256 input channels and 256 feature maps. Same padding is done to keep the size of the input and the output same. A kernel of size $8 \times 1$ extracts the features from the input. Stride is set to 1 for all the convolutional layers except for the first layer where stride is 2. Each convolutional layer is followed by a ReLU activation function and a dropout layer for regularization. Dropout probability is 0.1. Finally, after reordering the dimensions, two fully connected layers or linear layers are added at the end of the network. There are 512 input channels and 1,024 output channels in the first linear layer. In the second linear layer, there are 1,024 input channels and 73 output channels (number of grapheme tokens).

3.2. Deep Speech 2
The key to this architecture is recurrent layers which takes an input sequence and converts it into corresponding transcription [16]. More precisely, the whole architecture can be divided into three parts: two 2D convolutional layers, seven bidirectional GRU layers and one fully connected (FC) layer (Fig. 2). In the first two layers, temporal convolution is used in the time-and-frequency domain to capture the low-level features and make the network computationally less expensive. Each convolutional layer is followed by a batch norm layer to reduce internal covariate shift and then Hard tanh function as non-linearity. In the first convolutional layer, a $41 \times 11$ sized filter is used with stride size 2. The second layer has a filter with $21 \times 11$ dimensions and we stride the filter by 2 frames in the time domain and 1 frame in the frequency domain. $20 \times 5$ and $10 \times 5$ dimensional padding are done in the first two layers consecutively. Then we apply deep bidirectional GRU network with 7 hidden layers and 800 hidden units. Each recurrent layer is preceded by batch norm layer with $1e-05$ as epsilon value. At the end of the architecture, there is one more batch norm layer followed by a FC layer. There are 1,200 input features fully connected with 73 output features which provide outputs as Bangla graphemes or characters.

4. DECODING
We approach with two kinds of decoding methods. The first one is greedy decoding which finds the greedy best
path based on AM predictions. The second type of decoder we use is a beam search decoder. We incorporate an external N-gram language model to this decoder.

5. CORPUS DETAILS

5.1. Speech Corpus

We use Google Bangla speech corpus for our experiments [11]. Although it is mentioned in their paper that total duration of the corpus is 229 hours, but we found approximately 210 hours of correctly labeled speech data. Till now, it is the largest open-source corpus in Bangladeshi Bangla language. There are 508 native Bangladeshi speakers whose speeches are recorded to develop the corpus. There are 50,162 unique words in our speech corpus. We split the corpus into three sets: training, validation and test sets. Duration of training set is 170.26 hours, validation set is 18.76 hours and test set is 18.71 hours long. All the samples on the corpus are divided into three sets completely randomly to have a fair testing of how our systems perform. The utterances from the same speaker are included in the training and test sets. There are 18,924 samples in the test set.

5.2. Text Corpus

We build several word level N-gram LMs on a large text corpus. The file size of the text corpus is 9.7 gigabytes (GB). As Bangla is a Morphologically Rich Language (MRL), number of unique words can be much higher. Our text corpus has 10.6 million unique words. The total number of words in the text corpus is 602.5 million. There are 48.56 million lines of text in our corpus. All the words from the Google Bangla speech corpus are also found in our text corpus. The out-of-vocabulary (OOV) words, that are not included in the speech/text corpus, are handled by our lexicon-free acoustic model without any modification. This text corpus is built by web scrapping open-source websites—mainly Bangla newspaper portals [17] and Wikipedia pages. We use Kenlm language modeling toolkit to build the LM [18].

6. EXPERIMENTS AND RESULTS ANALYSIS

6.1. Experimental Setup

All of our experiments are done on a computer with 252 GB RAM, 48 intel Xeon CPUs and GeForce GTX 1080Ti graphics card. We run the training on 7 GPUs and decoding on 6 GPUs-each one of these GPUs is 11 GB. It takes us approximately 42 hours to train our CNN-based acoustic model and 70 hours for the DS2 model. However, DS2 model converges to the minimum WER possible within 50 epochs whereas CNN model requires more epochs to get to the best result.

We use wav2letter++ ASR research toolkit to train and evaluate the CNN model [19]. Table 1 shows the experimental parameters of our best performing CNN model and DS2 model.

6.2. Impact of Different Frame Sizes and Frame Strides

Speech is a non-stationary signal and so it is split into small segments or frames in which it can be assumed as stationary. We train 5 models with different frame sizes and strides using CNN. Each one of these models has 15 convolutional layers and all the other training parameters are also same to keep the fairness of the experiment. We observe that if frame size is 25 ms and stride is 15 ms, then it leads to better acoustic feature extraction for our dataset (Fig. 3).

6.3. Comparison between CNN-CTC and DS2 Models

Table 2 shows a comparison between RNN-based DS2 ASR system and deep CNN-based system. Both of these models are trained on the same training dataset and then evaluated using the same validation set and test set. Our best performing CNN model has 25 ms frame size and 15 ms stride size. Moreover, 20 convolutional layers are used while training to leverage deep convolutional network.
Figure 4 presents WERs and CERs for varied number of GRU layers, starting from 3 layers to 7 layers. Each of these layers has 512, 800 and 1,024 hidden units. WERs and CERs are calculated using both greedy and beam decoding approaches. We experiment with model complexity to ensure that our DS2 model doesn’t become overfitted and also we find out that model with 7 GRU layers and 800 hidden units yields the best result for this dataset.

Despite significantly less hours of training time and less number of trainable parameters (See Sect. 6.1), CNN-based model outperforms DS2 model achieving 36.12% WER whereas DS2 model obtains 40.91% WER. However, DS2 model is able to get slightly better character error rate than the CNN CTC model.

Table 2 also presents the results of the RNN-based DS2 system and our CNN-based system when an external large-scale 5-gram language model is integrated. DS2 model yields 24.08% WER and 11.09% CER with language model. This is a significant improvement in accuracy compared to the CTC-based greedy decoding approach.

Table 2 Comparison between CNN-CTC and Deep Speech 2 models.

| Model       | No LM | LM |
|-------------|-------|----|
|             | CER   | WER | CER  | WER |
| Deep Speech 2 | 13.13 | 40.91 | 11.09 | 24.08 |
| CNN CTC     | 13.39 | 36.12 | 8.42  | 13.93 |

Different LM weight, word score and beam width are chosen using grid search to get the best result. With our CNN-based acoustic model and the same 5-gram model, we manage to reduce it to 13.93%. We empirically show that CNN-based model outperforms the DS2 model by a large margin when LM is incorporated. In the next section, we go into detail about some additional LM-based decoding techniques implemented in our CNN-based ASR system (See Sect. 6.4).

6.4. Analysis of Language Model

Language model estimates the probability of a given sequence of words. Morphologically rich languages require more research on high-order $N$-gram LMs [20]. We build 3-gram, 4-gram and 5-gram LMs using the same large text corpus. In our experiment, 5-gram LM outperforms the 3-gram model by a significant margin (Fig. 5). Comparing the 5-gram LM with the 4-gram model, some improvement in accuracy can be observed.

We also experiment with different techniques of beam search decoding using the 5-gram LM. We decode with different beam sizes to see how it impacts the accuracy and decoding speed. As beam size increases, WER gradually decreases although time taken to decode also increases (Table 3). Language Model weight is also a key factor. If impact of LM is too low or high, we can’t get the optimal result (Fig. 6). We get the best accuracy using the value 3.75 as LM weight.

Another important element is LM lookahead or smearing of the LM [21]. During the first pass, when end
of a word or terminal node is reached, then word LM score is calculated. In our decoder, a heuristic approach is taken by calculating LM score early as it proceeds with tokens using unigram LM score.

While propagating score in the trie, maximum of children nodes scores and current node score can be taken (max smearing) or logadd operation can be done \([22]\). Logadd operation is defined as following:

\[
\text{logadd}(a, b) = \log(\exp(a) + \exp(b)) \tag{1}
\]

We achieve the best result when we do maximum smearing (Fig. 7).

### 6.5. Character-wise Error Analysis of Google Bangla Speech Corpus

We compute and analysis the character-wise error rates of Bangla characters including Bangla alphabets, numerical digits and some special characters using Google Bangla speech corpus. Table 4 shows the total number of occurrence of each character in the Google Bangla corpus along with pronunciation and error rate of each character calculated using both CNN-CTC and RNN-CTC based DS2 approaches. Error rate of each character is also examined with and without the LM. From Table 4, it can be seen that there is a strong correlation between the frequency of a character in the dataset and its error rate.

\( /k/\), \( /t/\), \( /n/\), \( /b/\), \( /m/\), \( /j/\), \( /l/\), \( /s/\), \( /\cdots /\) (for clustering two consonants), \( /a/\), \( /i/\) and \( /e/\) characters are found more than one hundred thousand times and their error rates are around 10%. On the contrary, \( /i/\), \( /u/\), \( /ri/\), \( /oy/\), \( /ow/\), \( /R/\), \( /R/\), \( /Y/\) and \( /\cdots H/\) (use as a diacritic) characters are found only less than one thousand times and these get almost 50% or more error rates without any LM applied. Among these characters, there is only one \( /R/\) found in the entire dataset and this one is included in our training set. So we cannot test the error rate of this character using our test set. In addition, \( /ow/\) alphabet is found only 7 times in our test set. CNN model cannot recognize this character at all whereas error rate of this character for DS2 model is 71.4%. We can also observe that the characters which are most frequent are generally better recognized by the DS2 than the CNN model.

\( /O/\), \( /a/\), \( /i/\), \( /i/\), \( /u/\), \( /\cdots /\) and \( /ow/\) alphabets are known as (Soroborno) or Bangla vowels. Among these alphabets, \( /i/\) and \( /u/\) are pronounced exactly the same as \( /i/\) and \( /\cdots /\) respectively but with longer duration. \( /i/\) and \( /\cdots /\) cannot be recognized well by both models. The frequencies of the ten numerical digits (\( /\cdots n/\), \( /\cdots e/\), \( /\cdots d/\), \( /\cdots t/\), \( /\cdots c/\), \( /\cdots r/\), \( /\cdots o/\), \( /\cdots y/\), \( /\cdots a/\) and \( /\cdots a/\) alphabets are known as (Soroborno) or Bangla vowels. Among these alphabets, \( /i/\) and \( /\cdots /\) are pronounced exactly the same as \( /i/\) and \( /\cdots /\) respectively but with longer duration. \( /i/\) and \( /\cdots /\) cannot be recognized well by both models. The frequencies of the ten numerical digits (\( /\cdots n/\), \( /\cdots e/\), \( /\cdots d/\), \( /\cdots t/\), \( /\cdots c/\), \( /\cdots r/\), \( /\cdots o/\), \( /\cdots y/\), \( /\cdots a/\) and \( /\cdots a/\)
and /\noY/) are not significant in this corpus and so accuracy of these digits are not well. After integrating LM, we see substantial improvement in reducing error rates of the characters. However, CNN model outperforms DS2 by a large margin with LM. We argue that the improved quality of the transcriptions and better WER from the greedy decoder of the CNN model make it easier for the LM to correct the existing errors and as a result, the CNN model performs superior to DS2 with LM. Both of the models do quite well in recognizing the spaces.

From this discussion, we can conclude that all the characters in the corpus are not well-balanced. Thus, a better corpus is required to facilitate Bangla ASR research.

Table 5 shows the transcriptions of two continuous speeches obtained from CNN-CTC and DS2 models. These speeches are not part of our corpus and recorded by an unseen speaker in a noisy environment. We can see that CNN model performs better than DS2. After applying our large 5-gram LM, we observe significantly less number of errors.
7. CONCLUSION AND FUTURE SCOPE

In this paper, we implement CNN-CTC based acoustic model and RNN-CTC based Deep Speech 2 model for Bangla LVCSR. We also investigate various large-scale $N$-gram LMs and observe that high-order $N$-gram model improves accuracy of the LVCSR for Bangla. We experiment with different decoding techniques with LM to see its impacts. This study shows that deep CNN-CTC model achieves 36.12% WER and RNN-CTC based Deep Speech 2 model obtains 40.91% WER on a specific benchmarked large corpus. With a 5-gram LM integrated to the CNN-CTC model, we reduce WER to 13.93% which is the state-of-the-art WER on this corpus. Our study on Google Bangla speech corpus suggests that a large-scale and well-balanced speech corpus is required and thus we are now developing a large corpus ensuring variability of speakers’ regional accented speech, grammar and vocabulary. Moreover, large high-order $N$-gram model is needed as well to improve Bangla LVCSR. More study about LM is also important as it can play a vital role in building a robust LVCSR system for a morphologically rich and under-resourced language — Bangla, i.e., the impact of the Neural Network (NN) based LM can be studied, because several researchers have found better impact of NN-based LM on morphologically rich and low-resource languages [23–25].

REFERENCES

[1] C. H. Lee, Advances in Chinese Spoken Language Processing (World Scientific, Singapore, 2007).
[2] A. Lüdeling and M. Kytö, Corpus Linguistics: An International Handbook, Vols. 1, 2 (De Gruyter Mouton, Berlin, 2008).
[3] B. Das, S. Mandal and P. Mitra, “Bengali speech corpus for continuous automatic speech recognition system,” Proc. Int. Conf. Speech Database and Assessments (Oriental COCOSDA), pp. 51–55 (2011).
[4] S. J. Young, The HTK Hidden Markov Model Toolkit: Design and Philosophy (Department of Engineering, University of Cambridge, United Kingdom, 1993).
[5] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouveia, P. Wolf and J. Woelfel, Sphinx-4: A Flexible Open Source Framework for Speech Recognition (Sun Microsystems, Inc., United States, 2004).
[6] M. A. A. Amin, M. T. Islam, S. Kibria and M. S. Rahman, “Continuous Bengali speech recognition based on deep neural network,” Proc. 2019 Int. Conf. Electrical, Computer and Communication Engineering (ECCE), pp. 1–6 (2019).
[7] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsy, G. Stemmer and K. Vesely, “The Kaldi speech recognition toolkit,” Proc. IEEE 2011 Workshop Automatic Speech Recognition and Understanding, IEEE Catalog No.: CFP11SRW-USB (2011).
[8] "larpal bengal bengali language pack," https://catalog.ldc.upenn.edu/LDC2016S08 (accessed 11 August 2020).
[9] S. H. Sumit, T. A. Muntasir, M. M. A. Zaman, R. N. Nandi and T. Sourov, “Noise robust end-to-end speech recognition for Bangla language,” Proc. 2018 Int. Conf. Bangla Speech and Language Processing (ICBSLP), pp. 1–5 (2018).
[10] S. Ahmed, N. Sadeq, S. S. Shubha, M. N. Islam, M. A. Adnan and M. Z. Islam, “Preparation of Bangla speech corpus from publicly available audio & text,” Proc. 12th Language Resources and Evaluation Conf., pp. 6586–6592 (2020).
[11] O. Kjartansson, S. Sarin, K. Pipatsrisawat, M. Jansche and L. Ha, “Crowd-sourced speech corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangadeshi Bengali,” Proc. 6th Int. Workshop Spoken Language Technologies for Under-Resourced Languages, pp. 52–55 (2018).
[12] S. Kibria, M. S. Rahman, M. R. Selim and M. Z. Iqbal, “Acoustic analysis of accent-specific pronunciation effect on Bangladeshi Bangla: A study on Sylheti accent,” Proc. 2018 Int. Conf. Bangla Speech and Language Processing (ICBSLP), pp. 1–4 (2018).
[13] S. Kibria, M. S. Rahman, M. R. Selim and M. Z. Iqbal, “Acoustic analysis of the speakers’ variability for regional accent-affected pronunciation in Bangladeshi Bangla: A study on Sylheti accent,” IEEE Access, 8, 35200–35221 (2020).
[14] M. N. Y. Ali, S. M. A. Al-Mamun, J. K. Das and A. M. Nurunnabi, “Morphological analysis of Bangla words for universal networking language,” Proc. 3rd Int. Conf. Digital Information Management, pp. 532–537 (2008).
[15] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingsbury. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Process. Mag., 29(6), pp. 82–97 (2012).
[16] D. Amodei, S. Anantharayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Olsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan and Z. Zhu, “Deep speech 2: End-to-end speech recognition in English and Mandarin,” Proc. Int. Conf. Machine Learning, pp. 173–182 (2016).
[17] A. Ahmad, M. R. Talha, M. R. Amin and F. Chowdhury, “Pipilika N-gram viewer: An efficient large scale N-gram model for Bengali,” Proc. Int. Conf. Bangla Speech and Language Processing (ICBSLP), pp. 1–5 (2018).
[18] K. Heafield, “Kenlm: Faster and smaller language model queries,” Proc. 6th Workshop Statistical Machine Translation, pp. 187–197 (2011).
[19] V. Pratap, A. Hannun, Q. Xu, J. Cui, J. Kahn, G. Synnaeve, V. Liptchinsky and R. Collobert, “wav2letter: The fastest open-source speech recognition system,” https://github.com/facebookresearch/wav2letter (accessed 11 November 2020).
[20] T. Hirsimaki, J. Pylkkonen and M. Kurimo, “Importance of high-order n-gram models in morph-based speech recognition,” IEEE Trans. Audio Speech Lang. Process., 17, 724–732 (2009).
[21] V. Steinbiss, B. H. Tran and H. Ney, “Improvements in beam search,” Proc. 3rd Int. Conf. Spoken Language Processing (ICSLP), pp. 1–5 (1994).
[22] R. Collobert, C. Puhrsch and G. Synnaeve, “Wav2letter: An end-to-end convnet-based speech recognition system,” arXiv preprint arXiv:1609.03193 (2016).
[23] A. Gandhe, F. Metze and I. Lane, “Neural network language models for low resource languages,” Proc. 15th Annu. Conf.
[24] T. Mikolov, J. Kopecky, L. Burget, O. Glembek and J. Černocký, “Neural network based language models for highly inflective languages,” Proc. 2009 IEEE Int. Conf. Acoust. Speech Signal Process., pp. 4725–4728 (2009).

[25] A. Mousa, H. K. Kuo, L. Mangu and H. Soltau, “Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic,” Proc. 2013 IEEE Int. Conf. Acoust. Speech Signal Process., pp. 8435–8439 (2013).