International Journal of Speech Technology (2021) 24:419–424
https://doi.org/10.1007/s10772-021-09811-5

Exploring end-to-end framework towards Khasi speech recognition system

Bronson Syiem[1] · L. Joyprakash Singh[1]

Received: 16 March 2020 / Accepted: 8 November 2020 / Published online: 27 January 2021

Abstract

Building a conventional automatic speech recognition (ASR) system based on hidden Markov model (HMM)/deep neural network (DNN) makes the system complex as it requires various modules such as acoustic, lexicon, linguistic resources, language models etc. particularly with the low resource languages. In contrast, End-to-End architecture has greatly simplifies the model building process by representing complex modules with a simple deep network and by replacing the use of linguistic resources with a data-driven learning techniques. In this paper, we present our prior work by exploring End-to-End (E2E) framework for Khasi speech recognition system and the novel extension towards the development of speech corpora for standard Khasi dialect. We implemented the proposed E2E model by using Nabu ASR toolkit. Additionally, three other models (monophone, triphone and hybrid DNN) were built. Comparing the results, significant improvement was achieved using the proposed method particularly with the connectionist temporal classification (CTC) with a character error rate (CER) of 5.04%.

Keywords Automatic speech recognition · Deep neural network · End-to-End · Hidden Markov model

1 Introduction

In the past decades, the Hidden Markov model (HMM) had been a widely used technique in the-state-of-the-art automatic speech recognition (ASR) system. A typical ASR system is factorized into several modules based on a probabilistic noisy channel model, including acoustic, lexicon and language models (Hori et al. 2017). Machine learning methods such as deep learning have powered dramatic improvements in acoustic and language models over the past decade (Hori et al. 2017). Current systems, however, rely heavily on scaffolding complicated legacy architectures which grew up around traditional techniques, including HMM, Gaussian Mixture model (GMM), deep neural networks (DNN), followed by sequence discriminative learning. We also need to develop a dictionary for pronunciation and a language model which involves linguistic awareness and text (Hori et al. 2017). One of the challenges of speech recognition is the wide range of speech and acoustic variability, which means that the modern ASR pipelines consist of numerous components, including complex feature extraction, acoustic models, language and pronunciation models, voice adaptation (Amodei et al. 2016). The design and tuning of these individual components makes it difficult to develop a new speech recognizer, particularly for a new language (Amodei et al. 2016).

An E2E speech recognition model usually represents a simple model that can be developed from scratch, and generally works directly on sentences, subwords, and characters/graphemes. This eliminates the need for a pronunciation lexicon and the entire explicit modeling of the phone and simplifies the decoding considerably (Zeyer et al. 2018; Watanabe 2017). Typically, E2E ASR approaches depend only on combined acoustic and language information without language knowledge, and train the system with a single algorithm. The approach therefore potentially enables ASR systems to be built without expert knowledge of the language (Zeyer et al. 2018). E2E speech recognition neural systems basically replace the HMM with a neural network that specifically distributes sequences (Zhang et al. 2016). There are two main types of E2E architectures of ASR:

✉ Bronson Syiem
bronzoe12@gmail.com

L. Joyprakash Singh
jplairen@gmail.com

[1] Electronics & Communication Engineering, NEHU, Shillong, Meghalaya 793022, India

attention-based approaches that uses an attention system to connect acoustic frames with known symbols and the other one is connectionist temporal classification (CTC) which uses Markov principles to solve sequential problems effectively by dynamic programming (Zeyer et al. 2018).

In this study, we performed our experiment by incorporating E2E framework for Khasi speech recognition and a novel extension towards development of speech database for standard Khasi dialect which is one of the major dialect spoken in the state of Meghalaya, a state in North East India.

1.1 Listen, Attend and Spell

Listen, attend and spell (LAS) is a sequence to sequence attention-based model that learns to transcribe an audio sequence to a word sequence, one character at a time (Zhang et al. 2017). LAS consists of a recurrent neural network (RNN) encoder and decoder called a listener and a speller (Fig. 1). The Listener is a pyramidal structured bi-directional long short-term memory (pBLSTM) that takes sequence of low level time-frequency acoustic features, $\mathbf{x}(x = x^1, x^2, x^3, ..., x^T)$, as input and encode the same to high level hidden features representation $\mathbf{h}(h = h^1, h^2, h^3, ..., h^U)$ with U ≤ T (Shan et al. 2019). The encoder output can be expressed as in Eq. (1) (Shan et al. 2018).

$$h = Listen(x) \tag{1}$$

The speller (consist Attend and Spell modules) is an AttendAndSpell-based decoder that converts these higher-level features into output expressions by specifying a distribution of probability over character sequences conditioned on preceding outputs (Chan et al. 2016). The Attend module decides which encoder features should be used to predict the next output symbol, resulting in a context vector. The Spell module takes the context vector resulted from Attend module and an embedding of the previous prediction to generate a prediction of the next output (Shan et al. 2018). Mathematically, the output of the speller can be represented using Eq. (2) (Shan et al. 2018).

$$P\left(y_i|x,\ y_{<i}\right) = AttendAndSpell(y, h) \tag{2}$$

where **y** represents the predicted character conditioned on preceding character.

1.2 Connectionist temporal classification

CTC is a network that enables an RNN to be trained for sequence transcription tasks without needing prior alignment that monotonically maps an input sequence to a shorter output sequence (Gelabert et al. 2017; Hori et al. 2017). Unlike other hybrid system, this approach is not trained using frame-level labels rather it uses CTC objective function to learn the alignment between speech frames and their labels (e.g. phonemes, characters etc.) (Miao et al. 2015). It consist of deep BLSTM (DBLSTM) as encoder and a DNN-based decoder (Fig. 2). In contrast to LAS approach, CTC does not make any prediction conditioned on all the previous predictions. Instead, it assumes that the output labels are conditionally independent from each others at different time steps (Gelabert et al. 2017; Sumit et al. 2018). In CTC, it uses a blank symbol that expands the length, **L**-sequence, of the target symbols to the length **T**-sequence (Kurata and Audhkhasi 2018). There are as many possible alignments for a given transcription sequence as there are different ways of separating the labels with blanks. If '_' to denote blanks, the alignments (_, a, _, b, c, _) and (a, _, b, _, _, c) both correspond to the same transcription (a, b, c). Furthermore, if the same label appears on consecutive time-steps

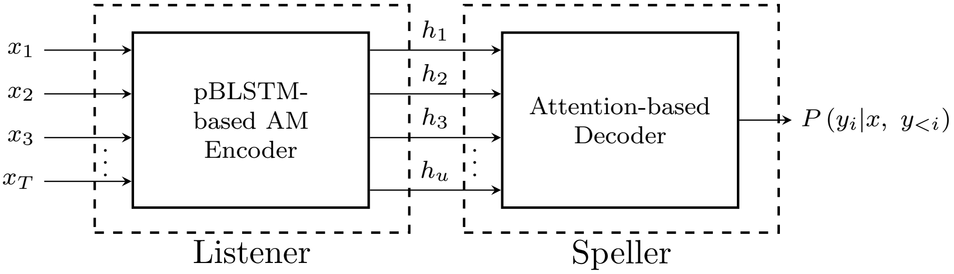

Fig. 1 Block diagram of LAS network

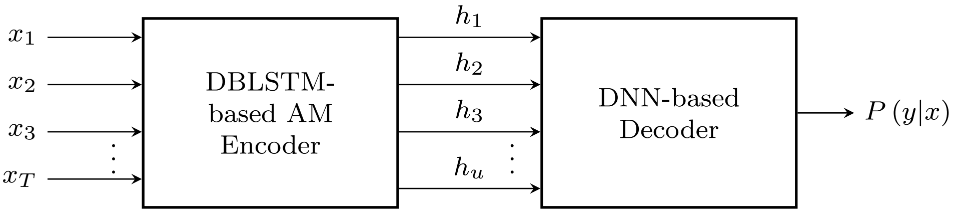

Fig. 2 Block diagram of CTC networ

in an alignment, the repeats are removed. Therefore (a, b, b, b, c, c) and (a, _, b, _, c, c) also correspond to (a, b, c). The Network decides intuitively whether to emit any label or no label at every time-step. Considering these decisions together, a distribution over alignments between input and target sequences is defined. CTC then uses a forward-backward mechanism to sum over all possible alignments and determine the normalized probability of the target sequence given the input sequence (Gelabert et al. 2017). Mathematically, the encoder and the decoder outputs of CTC network can be represented as shown in Eqs. (1) and (2) respectively.

P(y|x) represents the alignment of character **y** with respect to the acoustic feature **x**. Thus

$$h = Encode(x) \tag{3}$$

$$P(y|x) = Decode(x) \tag{4}$$

2 Related work

Our work is inspired by several published works. (Bachate et al. 2019) described various approaches towards the development of ASR system and the needs of ASR-based technologies. Dario at el. performed E2E speech recognition for two different languages, their result shows that this approach can improve recognition as well as can handle diverse variety of speech such as accents and different languages (Amodei et al. 2016). E2E-based speech recognition system outperforms the existing methods in different challenging scenarios such as clear, conversation and noisy speech (Hannun et al. 2014). Although, DNN model can achieve tremendous improvement in the development of robust ASR system. However, it remains a challenging task (such as, resources, training stages, expert etc.). Such issues can be overcome using CTC (Miao et al. 2015). Significant reduction on word error rate (WER) were observed using E2E speech recognition models with different data base (Li et al. 2019). Attention-based model can improve the performance over another E2E approach. However, it provides poor results on noisy data (Kim et al. 2017). Performing E2E framework on two separate databases (LibriSpeech and SwichBoard) shows much improvement in the recognition (Park et al. 2019). CTC approach can be used for labeling unsegmented sequences that makes it feasible to train an E2E speech recognition system (Zhang et al. 2016).

3 Experimental set up

We carried out our experiment on Ubuntu 18.04 platform, using Kaldi and Nabu ASR toolkit to verify the performance of the proposed method. Kaldi is an open-source ASR toolkit, the benefit of Kaldi-based application on speech recognition creates high-quality lattices and is quick enough for real-time recognition (Guglani and Mishra 2018). A recipe in Kaldi is a series of steps describing scripts that will allow a user to create a recognizer for some speech data base and to encourage the knowledge and to make the speech recognition software available to programmers and scientists worldwide (Guglani and Mishra 2018). Nabu is an ASR toolkit for E2E networks built on Tensor-flow v1.8 for implementing LAS and CTC speech recognition (Gelabert et al. 2017). Nabu works in various stages: preparation of data, training, and finally testing and decoding. For a specific model and database, each of these stages uses a separate recipe. The recipe contains all component configuration files and defines all the parameters needed for the database and the model (Renkens 2019).

3.1 Speech corpora

Our speech database consist of 30.12 (12,050 wave files) hours read speech data at the sampling rate of 16 kHz from 241 native speakers. Each speaker was given 50 sentences to read. Recordings of speech were made using Zoom H4n handy recorder in a laboratory condition. The duration of speech files are within 3–19 s approximately.

3.2 Text corpora

For this experiment, a total of 12,050 transcribed labeled (consist 119k Khasi words) files corresponding to the speech files were used. These labeled files were prepared as per the requirement of the ASR tool kits. We have used a total of 31 symbols including "_" and "sil" to make the label files. Table 1 shows the arrangement of labels of few Khasi words, for example, used in the experiment. In our text corpora, symbol "_" is used to separate the words.

Table 1 Illustration of labels used for the experiment

Khasi words	Meaning in English	Labels
Blei	God	b l e i
khublei	thank	kh u b l e i
kumno	how	k u m n o
kumno phi long	how are you	k u m n o _ ph i _ l o ng

We performed our experiments with speech data containing 30.12 h for training and 3.75 h for testing. We extract 123-dimensional Melscale filter-banks coefficients with first and second order derivatives by applying Hamming window of size 25 ms with a frame shift 10 ms normalized with mean and variance for each speaker. These filter-bank features were used as input to the proposed E2E models. Building E2E-based speech recognition system using Nabu ASR toolkit involves various stages. In the first stage, all the data required for both training and testing (feature computation, target normalization etc.) were prepared. Before running database preparation, database.conf file was created in the recipe directory based on the already existing database.cfg file and all the paths were filled. In the training stage (second stage) the model is trained to minimize the loss function. In order to adjust the learning rate, different configuration files (e.g. model.cfg, trainer.cfg and validation evaluator.cfg) were used. This is followed by the testing stage where evaluation of the model performance is carried out. The last stage is the decoding stage where the model is used to decode the test set and to select the best list. In this stage, recognizer.cfg file is used for modification of the model used for decoding.

LAS' Encoder is a two pBLSTM layers with one non-pyramidal layer at the end of it. Each hidden layer consists of 256 hidden units. The decoder is a Speller-based system having two hidden layers with 128 hidden units in each layer. CTC model consists of two DBLSTM-based layers in the encoder stage with 256 hidden units in each layer. Decoder contains one DNN-based layer with 256 hidden units. We set initial learning rate to 0.001 and to reduce over fitting, we used dropout and set the value to 0.5 throughout all our E2E experiments. The performances are measured by character error rate (CER).

In addition to E2E models, we have also developed three other models (i.e. monophone-based GMM-HMM, Triphone-based GMM-HMM and hybrid HMM-DNN) to compare the results obtained with the proposed methods. The models were built using Kaldi ASR toolkit with static Mel frequency cepstral coefficient (MFCC) along with delta and delta-delta coefficients as input features. Context-independent Monophone-based HMM-GMM model consist of five states HMM, where each state is represented by having a single Gaussian mixture with mean, variance and mixture weights. Similarly, context-dependent/ tied state Triphone acoustic model was developed with five state HMM. DNN model used in this experiment is a feed-forward neural network with 5 hidden layers, each layer consists 256 hidden nodes.

4 Results and discussion

As discussed in Sect. 3, Khasi speech recognition system using Kaldi and Nabu ASR toolkits in the ubuntu platform was developed. A comparison of results obtained with the proposed and the conventional methods is shown in Table 2. Using classical methods (monophone and triphone), the experimental outcome showed poor efficiency particularly with monophone model and this may be due to the existing insufficient variation of phones with respect to the left and the right context (Guglani and Mishra 2018). As seen from (Fig. 3) some improvement can be seen using context-dependent triphone model. In contrast to monophone model, this model can capture the varying articulation that a phone is subject to when it is realized to different surrounding phonetic contexts. However, the result obtained was not satisfactory. The reason might be due to the labeled used in the experiment where character sequence was used rather than word sequence. Re-scoring the existing methods with recurrent neural network based language model (RNNLM) also did not contribute much towards the enhancement of performances. Further observation for improvement in performances was made by incorporating hybrid HMM-DNN system with different hidden layer sizes in addition to RNNLM. Though much improvement was observed, the outcome of this method too was not satisfactory. Significant improvement was observed with the proposed methods, particularly with the CTC approach. The proposed methods outperformed the other models, this because, E2E models are trained as frame level classifier which means separate training target is used for every frame of the speech signal (Gelabert et al. 2017). Moreover, it was found that the experiment produced no further progress using the LAS method regardless of the parameter shift as oppose to HMM-DNN

Table 2 Comparison of CER from different models

Model	CER (%)
Monophone	38.28
Triphone	25.38
DNN	16.34
Monophone with RNNLM	35.42
Triphone with RNNLM	21.53
DNN with RNNLM	13.03
LAS	18.55
CTC	5.04

and CTC. This can be due to the presence of noise in our sampled data as stated in (Kim et al. 2017). Furthermore, it was observed that in term of computational time complexity, although E2E models took more time for training as compared to the other three models, yet decoding speed was more high. Table 3 shows one short selected recognized sentence evaluated from different models with respect to ground truth (GT).

5 Conclusion

In this experiment, we explored E2E-based speech recognition system on standard Khasi dialect. Using classical methods, we observed that there is no much improvement even with the use of RNN-based language model. However, proposed methods show significant improvement in the recognition performance. From the experiment, it was found that there is computational time complexity while training the E2E models unlike DNN and the other two classical models. However, E2E models provided promising results particularly with CTC. Furthermore it was found that E2E models takes less decoding time as compared to the other models developed. As for future plan, we may explore other machine learning approaches with more speech data and incorporate language modeling.

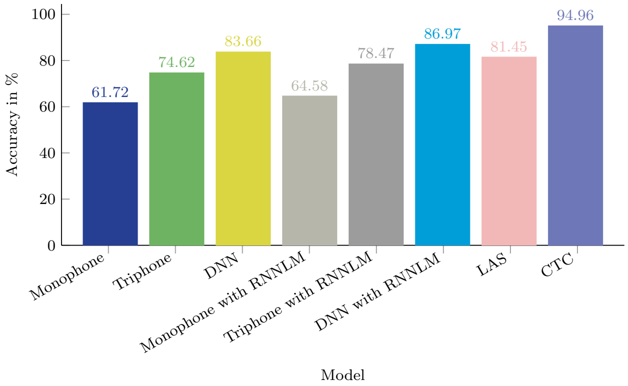

Fig. 3 Recognition accuracy obtained from different models

Table 3 Character sequence of a selected sentence evaluated from different models (* represent deleted characters)

GT	h	a	b	a	_	i	a	_	k	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	r	w	a	i	_	ng	a	_	k	y	n	u	d
Monophone	h	a	b	a	_	i	*	_	k	i	_	ng	*	_	s	*	a	m	_	b	a	n	_	*	w	a	*	_	ng	a	_	k	y	n	u	t
Monophone with RNNLM	h	a	b	a	_	i	*	_	k	i	_	*	*	_	s	*	a	m	_	b	a	n	_	*	w	a	*	_	*	n	_	k	a	m	u	*
Triphone	h	a	b	a	_	i	a	_	s	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	*	w	a	i	_	*	a	_	k	y	n	u	*
Triphone with RNNLM	h	a	b	a	_	i	a	_	k	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	*	w	a	i	_	*	a	_	k	y	n	u	d
DNN	h	a	b	a	_	i	*	_	k	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	*	w	a	i	_	*	a	_	k	y	n	u	d
DNN with RNNLM	h	a	b	a	_	i	a	_	k	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	*	w	a	i	_	n	a	_	k	y	n	u	d
LAS	h	a	b	a	_	i	a	_	k	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	r	w	a	i	_	ng	a	_	k	y	n	u	d
CTC	h	a	b	a	_	i	a	_	k	i	_	ng	a	_	s	d	a	ng	_	b	a	n	_	r	w	a	i	_	ng	a	_	k	y	n	u	d

References

Amodei, D., et al. (2016). Deep speech 2: End-to-End speech recognition in English and Mandarin. In *Proceedings of the 33rd international conference on machine learning* (Vol. 48, pp. 173–182).

Bachate, R. P., & Sharma, A. (2019). Automatic speech recognition systems for regional languages in India. *International Journal of Recent Technology and Engineering*, *8*, 585–592.

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2016.7472621.

Escur i Gelabert, J. (2017). Exploring automatic speech recognition with TensorFlow (pp. 1–36). *Degree thesis*.

Guglani, J., & Mishra, A. N. (2018). Continuous Punjabi speech recognition model based on Kaldi ASR toolkit. *International Journal of Speech Technology*, *21*, 211–216.

Hannun, A., et al. (2014). Deep speech: Scaling up End-to-End speech recognition (pp. 1–12). arxiv.org/abs/1412.5567.

Hori, T., Watanabe, S., Zhang, Y., & Chan, W. (2017). Advances in joint CTC-attention based End-to-End speech recognition with a deep CNN encoder and RNN-LM. *Interspeech*. https://doi.org/10.21437/Interspeech.2017-1296.

Kim, S., Hori, T., & Watanabe, S. (2017). Joint CTC-attention based End-to-End speech recognition using multi-task learning. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2017.7953075,4835-4839.

Kurata, G., & Audhkhasi, K. (2018). Improved knowledge distillation from bi-directional to uni-directional LSTM CTC for End-to-End speech recognition. *IEEE Spoken Language Technology Workshop (SLT)*. https://doi.org/10.1109/SLT.2018.8639629.

Li, J., et al. (2019). Jasper: An End-to-End convolutional neural acoustic model. *Interspeech*. https://doi.org/10.21437/Interspeech.2019-1819.

Miao, Y., Gowayyed, M., & Metze, F. (2015). EESEN: End-to-End speech recognition using deep RNN models and WFST-based decoding. *IEEE Workshop on Automatic Speech Recognition and Understanding*. https://doi.org/10.1109/ASRU.2015.7404790.

Park, D. S., et al. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. *Interspeech*. https://doi.org/10.21437/Interspeech.2019-2680.

Renkens, V. Retrieved November 21, 2019, from https://www.github.com/vrenkens/nabu.

Shan, C., et al. (2019). Investigating End-to-End speech recognition for Mandarin-English code-switching. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2019.8682850.

Shan, C., Zhang, J., Wang, Y., & Xie, L. (2018). Attention-based End-to-End speech recognition on voice search. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2018.8462492.

Sumit, S. H., Al Muntasir, T., Zaman, M. A., Nandi, R. N., & Sourov, T. (2018). Noise Robust End-to-End speech recognition for Bangla language. *International Conference on Bangla Speech and Language Processing (ICBSLP)*. https://doi.org/10.1109/ICBSLP.2018.8554871.

Watanabe, S. (2017). Hybrid CTC/attention architecture for End-to-End speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, *11*(8), 1240–1253.

Zeyer, A., Irie, K., Schluter, R., & Ney, H. (2018). Improved training of End-to-End attention models for speech recognition. *Interspeech*. https://doi.org/10.21437/Interspeech.2018-1616.

Zhang, Y., et al. (2016). Towards End-to-End speech recognition with deep convolutional neural networks. *International Conference on Intelligent Robotics and Applications*. https://doi.org/10.21437/Interspeech.2016-1446.

Zhang, Y., Chan, W., & Jaitly, N. (2017). Very deep convolutional networks for End-to-End speech recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2017.7953077.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.