Large Vocabulary Continuous Speech Recognition for Nepali Language

Elina Baral and Sagar Shrestha
Paaila Technology Private Limited, Lalitpur, Nepal
Email: elinabaral53@gmail.com, sagar@paailatechnology.com

***Abstract*—Speech Recognition is a widely studied topic for high-resource languages like English and Mandarin. A plethora of publications exist that study the performance of several recognition methods for these languages. However differences in phonetics, accent, language model, etc between any two different languages demand for a study of speech recognition methodologies and components separately for each language. In this paper, we present a comparative study of popular speech recognition methods for Nepali, a low-resource Indo-Aryan language. We describe our approach to building the phonetic dictionary and present our findings for DNN and GMM based techniques with speaker adaptation on 50K vocabulary speech recognition task.**

***Index Terms*—Nepali speech recognition, automatic speech recognition, Nepali speech processing, Nepali phonetic dictionary**

I. Introduction

Most modern speech recognition systems, with the exception of end-to-end models, consist of three main components, namely – acoustic model, lexicon and language model. Time-domain samples of speech signal are divided into overlapping windows of certain time period. The set of samples contained in a window is called a frame. A preprocessing step extracts a feature vector that captures phonetic characteristics for each frame. Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC) and Perceptual Linear Prediction (PLP) are the most widely used feature extraction methods in speech recognition. The job of the acoustic model is to transform this sequence of feature vectors into a sequence of phones that represents the utterance. Most commonly applied technique for speech recognition is the Hidden Markov Model (HMM). In HMM, a phone, or a triphone is represented by a state in the usual Markov model that cannot be directly observed. A mixture of gaussians can be used to model each state of the HMM. Such a model is called a Gaussian Mixture Model (GMM) which was the dominating method for four decades until Deep Neural Networks (DNN) took over. DNNs have shown to outperform GMM on various benchmarks due to its ability to learn models for data lying on or near non-linear manifold [1]. A basic block diagram of speech recognition system is shown in Fig. 1.

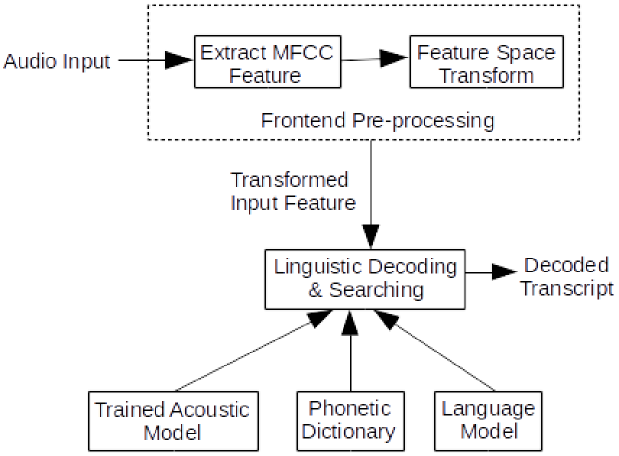

Figure 1. Simplified block diagram of automatic speech recognition system.

Performance of the model can be significantly increased by applying speaker adaptation techniques in feature space, called normalization and/or model space, known as adaptation. In speech recognition, a speaker is a general concept for different signal conditions that accounts for sources of variability in input features, thus affecting the model's performance. Normalization modifies feature vector to better fit the trained model while adaptation techniques modifies the trained model parameters to better fit the test speaker feature vector. Commonly used normalization techniques include Cepstral Mean and Variance Normalization (CMVN) [2], Vocal Tract Length Normalization (VTLN) [3], feature based Maximum LIkelihood Linear Regression (fMLLR). Various normalization techniques have been studied in [4]. Model space transformation includes affine transformation of model parameters using Maximum Likelihood Linear Regression (MLLR) [5], Maximum A Posteriori (MAP) estimation [6] and eigen voices. A survey of adaptation techniques can be found in [7].

Moreover, basic Maximum Likelihood training can be improved by the use of Maximum Mutual Information Estimation (MMIE). Inter-class variability can also be improved by Linear Discriminant Analysis (LDA) that transforms feature vectors to components along the axis where the gaussians can be better discriminated.

In this paper, many of the aforementioned approaches and their combinations have been evaluated for GMM models and DNN models. This paper has been organized into four major sections. Section II studies phonetic

Manuscript received July 16, 2020; revised November 22, 2020.

doi: 10.18178/ijsps.8.4.68-73

characteristics of Nepali language and builds up Nepali lexicon. Section III describes the GMM-HMM and DNN-HMM acoustic modeling techniques along with training methodologies. Section IV describes several speaker adaptation and sequence discriminative methodologies used for performance improvement. Section V describes all the experimental setup for all the models. Section VI tabulates results and presents a discussion on them. Section VI provides conclusions regarding the experiments and methodology. All the experiments were carried out in Kaldi toolkit [8]. Training scripts were adapted from the example scripts found in the toolkit.

II. Nepali Phonology

Nepali language has 11 distinct vowels (6 oral and 5 nasal) and 33 consonants. IPA tables for the vowels and consonants are shown in Tables I and II respectively. For computational purposes, these IPA symbols were represented by APRABET like characters. Since no prior work was available on LVCSR for Nepali language, phonetic dictionary was built from scratch. Nepali, being a phonetically written language, can be transcribed by rules in most of the cases. A grapheme to phoneme converter was built based on letter to phone correspondences and schwa deletion rules.

Finally, the most common 50,000 words were selected from the General Corpus Nepali Monolingual written corpus [10] which consists of 1,400,000 words collected opportunistically from various sources such as the internet webs, newspapers, books, publishers and authors.

III. Acoustic Modeling

A. Hidden Markov Model

The most common modeling to speech signal is the Hidden Markov Model (HMM). HMM treats a speech signal as composed of a sequence of elementary units called phones. The transition between adjacent phones is assumed to follow the Markovian process, i.e. the current state holds all the information about the entire history of the process. To account for the co-articulation phenomenon in speech signal, i.e. the dependence of speech features on the phone immediately before and after the current phone, tri-phone is usually considered a state of the HMM. The parameters of a HMM is the state transition probability and the output probability density. For a given HMM with state transition between state i and state j $\lambda = (A, B)$ where $A = a_{11}\ldots a_{ij}\ldots a_{NN}$ is the transition probability matrix and $B = b_i(o_t)$ is a sequence of observation likelihoods. Estimation of the HMM parameters is done through the Baum-Welch algorithm which is a special case of Expectation-Maximization Algorithm.

B. Gaussian Mixture Model

The emission probability density of the HMM states can be modeled using a mixture of gaussians. This model is termed as Gaussian Mixture Model (GMM). GMM is a sum of a number of d-dimensional gaussians which represents the probability density of the corresponding phone. Here, d is the dimension of the feature vector. For a GMM based HMM, estimating the emission probability translates to finding mean vector μ and covariance matrix Σ for the mixture of gaussians.

C. Deep Neural Networks

Deep Neural Networks have been proved to learn much better models of data than GMM models [1]. The reason lies in the strength of DNNs to learn representations of data points that lie on or near a non linear manifold. DNNs with many layers of hidden units and a very large output layer needed to accomodate all tied triphone states of the HMM. As such, DNNs essentially become multiclass classifier that takes as input the feature vector and outputs at each output unit the probability p_j of the given data point being in the class y_j that corresponds to the triphone state j.

D. WFST Framework

Speech recognition stack contains an acoustic model, lexicon and a language model. Each component can be efficiently modeled by using a Weighted Finite State Transducer (WFST). Composition of these components results in a big graph of HMM states that encodes all the information about lexicon and grammar or language model. More concretely, let G denote the grammar transducer that maps from word sequence to word sequence i.e. input to the state of transducer is a word and the output is the same word weighted by its probability of occurrence in the given context. Let L denote the pronunciation lexicon that takes in phone sequence and outputs a sequence of words, C be the context dependency transducer that converts context independent phones to context dependent and H denote the HMM set which is the closure of union of individual HMM. The integrated transducer results from the composition and determinization of the compositions as in (1).

$$\mathrm{N} = \det(\mathrm{H} \circ \det(\mathrm{C} \circ \det(\mathrm{L} \circ \mathrm{G}))) \qquad (1)$$

TABLE I. Consonants in Nepali Language [9]

	Bilabial		Dental		Alveolar		Retroflex		Palatal	Velar		Glottal
Nasal	m				n					ŋ		
Stop	p pʰ	b bʱ	t tʰ	d dʱ	tʃ tʃʰ	dʒ dʒʱ	t tʰ	d dʱ		k kʰ	g gʱ	
Fricative					s							ɦ
Rhotic					r							
Approximant	(w)				l				(j)			

TABLE II. VOWELS IN NEPALI LANGUAGE [9]

	Front	Central	Back
High	ɪ ĩ		u ũ
Close-mid	e ẽ		o
Open-mid			ʌ ʌ̃
Open		a ã	

IV. MODEL AND FEATURE SPACE IMPROVEMENTS

A. Speaker Adaptation

1) Maximum likelihood linear regression

There has been an extensive use of linear transforms for both training and adaptation of HMM based speech recognition systems. Linear transforms can be used for applications such as decorrelation of feature vectors and constrained adaptation of acoustic models to speakers. Maximum Likelihood Linear Regression uses linear transformation on the model parameters, means and variances, to adapt to a given speaker [5]. When applying MLLR, new mean vectors $\hat{\mu}$ and covariance matrices $\hat{\Sigma}$ are calculated, which can be implemented with either common global parameters or individual parameters for each state.

$$\mu = \mathbf{W}\mu + \mathbf{b} = \mathbf{A}\xi \tag{2}$$

where $\boldsymbol{W}$ is transformation matrix applied to mean vector, b the bias vector, $A = [Wb], \xi = [\mu\ \mathbf{1}]$.

$$\hat{\Sigma} = \mathbf{LBL}^T \tag{3}$$

where $L$is Choleski factor of Σ.

$$\hat{\Sigma} = \mathbf{B}\Sigma\mathbf{B}^T \tag{4}$$

And in both cases B is the transformation matrix to be obtained.

2) Speaker adaptive training

Speaker Adaptive Training (SAT) annihilates the inter-speaker variability and phonetic variation of the training population. Given a set of B speakers and their corresponding adaptation cepstra X_i for $1 \leq i \leq B$, SAT optimizes the maximum likelihood criterion on a per speaker basis as:

$$\arg\max_{\Theta, C_i} \prod_{i=1}^{B} p(C_i(\mathrm{X_i}) \mid \Theta) \tag{5}$$

where the individual speaker-dependent transforms C_i and the model parameters $\Theta = (\mu_1, \ldots, \mu_N, \Sigma_i, \ldots, \Sigma_N$ are jointly estimated. This optimization is done in a two step process, first estimating transforms C_i and second, retraining the model Θ. Such a two step process is iterated several times in E-M manner.

3) Feature-based MLLR

Feature-based MLLR (fMLLR) [13] which is also known as constrained MLLR applied to feature space applies adaptation transform to input feature vectors rather than model parameters . It is used to normalize features to better fit the speaker. It has proved to be highly effective as a method for unsupervised adaptation to a new speaker or environment. Following affine transform is applied to the feature vector.

$$\hat{\mathbf{x}} = \mathbf{Ax} + \mathbf{b} \tag{6}$$

where x is the feature vector, A is the transformation matrix and b is the bias vector.

$$\mathbf{x} = \mathbf{W}\xi \tag{7}$$

where $W = [A\ \ b]$ and $\xi = [b\ 1]$.

B. Other Optimization Techniques

1) Linear discriminant analysis

Linear Discriminant Analysis is used as dimensionality reduction technique with good class-separability and reduce computational costs. LDA projects a feature space (n-dimensional) onto a smaller subspace k (where $k \leq n - 1$) while maintaining the class-discriminatory information by computing eigenvectors from data and collecting them in scatter-matrices (i.e., in-between-class scatter matrix and within-class scatter matrix). The idea is to find a projection of the data where the variance between the classes is large compared to the variance within the classes. Under assumptions of Gaussian class distribution and a common with-in class covariance matrix this can be stated formally as finding a projection matrix θ that maximizes the quotient

$$J(\theta) = det(\theta\Sigma_b\theta^T)/det(\theta\Sigma_w\theta^T) \tag{8}$$

where Σ_b is the between-class covariance matrix and Σ_w is the common within-class covariance matrix.

2) Delta, delta-delta

A common method for extracting information about such transitions is to determine the first difference of signal features, known as the delta of a feature and the second difference is known as the delta-delta. To calculate the delta coefficients, the following formula is used:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{9}$$

where d_t is a delta coefficient, from frame t computed in terms of the static coefficients to c_{t-N}. A typical value for N is 2. Delta-Delta coefficients are calculated in the same way as detlas. Speech recognition systems conventionally append delta and double-delta cepstral features to static cepstral features. Delta-cepstral features capture dynamic speech information and improve ASR recognition accuracy, they are not robust to noise and reverberation.

3) I-vectors

I-vectors is used for low-dimensional representation of the input speech signal that captures the speaker- and channel-dependent variations [14]. The total variability matrix is composed of speaker and session variability simultaneously. Each factor of the i-vectors control the Eigen dimension of this matrix. The i-vector computation can be viewed as a probabilistic compression used to reduce speech-session super-vectors dimension. The speaker and channel dependent supervector M is projected into the total variability space as follows

$$M = m + Tw \tag{10}$$

where m is the mean super-vector of Universal Background Model (UBM), T is the total variability matrix and w is the resulting i-vector.

4) *Sequence discriminative training*

Unlike Maximum Likelihood Estimation (MLE) that tries to model the density of HMM state, discriminative training approaches speech recognition as a sequence classification problem by directly trying to maximize the probability of output class label given the acoustic input. Maximum Mutual Information Estimation (MMIE) and Minimum Phone Error (MPE) training criterion are described below. MMIE aims to directly maximize probability of word sequence given the observations [15]. Let $O_u = o_{u1}, \dots, o_{uTu}$ be the sequence of observations and W_u be the sequence of words corresponding to utterance u, then the MMI criterion is given by:

$$F_{MMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^k P(W_u)}{\sum_W p(\mathbf{O}_u|S_u)^k P(W)} \tag{11}$$

where $S_u = \{s_{u1}, \dots., s_{uTu}$ is the sequence of HMM states corresponding to W_u and k is the acoustic scaling factor. Denominator gives the total likelihood of the data given all possible word sequences. Computing of the denominator is done by generating latices, and summing over all the words in the lattice.

While maximizing MMIE, objective function minimizes the expected sentence error. State level Minimum Bayes Risk (sMBR) and Minimum Phone Error (MPE) work on state level and phone level respectively. They are designed to miniize error corresponding to the state or phone labels.

$$F_{MMI} = \sum_u \log \frac{\sum_W p(\mathbf{O}_u|S)^k P(W) A(W,W_u)}{\sum_{W'} p(\mathbf{O}_u|S_u)^k P(W')} \tag{12}$$

where $A(W, W_u)$ is the phone or state transcription accuracy of sentence W given reference sentence W_u.

V. Experiments

The experiment was conducted in two stages. First, a baseline GMM-HMM model was selected from models built with different speaker adaptation and optimization techniques. Second a DNN model was trained using the labels generated by the baseline GMM-HMM model for all training data. the 48k vocabulary dictionary created from the aforementioned process was used as the lexicon. Open sourced data found on [16] appended with nearly 60 hours of read speech data collected at Paaila Technology [17] was used to build the speech corpus. For training the baseline model, only the first source was used that consisted of nearly 150 hours of speech data comprising 158k utterances. The corpus was split into training and test set in 8:2 ratio. A subset of the data was created for evaluation of the performance of the model on varying data size. The subset consisted of 80 hours of speech data.

A. *Baseline Model*

Using the 3-gram language model, first a simple mono-phone model was trained to create alignments of phones with corresponding labels. This is necessary because acoustic models require phones to frame correspondence. Training utterances usually consist of a sequence of phones and a sequence of frames whose lengths are not equal. Better alignments are generated by cascading tri-phone models with larger number of parameters (number of tri-phone states and number of gaussians) for the ultimate baseline model.

Several models were created using different training methodologies. Feature and model space transformations were applied using Delta-Delta, MLLT and LDA. Using raw MFCC features a triphone model was trained using alignments generated by a monophone model. Using alignments created by the triphone model, first, a triphone model with 5500 tied-states and 90000 gaussians was trained using delta-delta features. In the second experiment, features transformed with LDA and MLLT were used to train a second model referred to as LDA-MLLT in this paper. And lastly, a combination of speaker adaptive training with feature transformation using Delta-LDA-MLLT was used to train the final model, referred to as SAT-LDA-MLLT. Each training was carried out in the two subsets of data. The Baseline Model was obtained by applying speaker adaptation via SAT with a 40-dimensional feature vector obtained from Delta-Delta and transformed with LDA.

B. *DNN Model*

The baseline GMM-HMM model produces labels for each frame of the input speech vector to the corresponding state in the HMM. This labeled frame is used as a training example to train the DNN model which will be able to predict the state of HMM given the input feature vector. Here we train three distinct models whose training recipes are available in the Kaldi toolkit. We call them DNN-sMBR [18], DNN-Pnorm [19], and DNN-TDNN-LSTM [20] respectively based on the architecture and training methodologies for these models. The three models are described as follows.

1) *DNN-sMBR*

For this model, a 40-dimensional feature is extracted. The features include MFCC-LDA-MLLT-fMLLR with CMN. The features are used to pre-train a Deep Belief Network using Contrastive Divergence with 1-step of Markov Chain Monte Carlo sampling (CD-1) [21]. Then a DNN is trained to classify frames into triphone-states using mini-batch Stochastic Gradient Descent (SGD). Finally sMBR sequence-discriminative training with input dimension of 440, output dimension of 2816, starting learning rate of 0.008 and no hidden layers is done to train the neural network in order to jointly optimize for the whole sentence.

2) *DNN-Pnorm*

This model is basically a feed forward neural network with pnorm non-linearity as the activation function. For this model, same 40-dimensional feature vector, i.e. MFCC (spliced)-LDA-MLLT-fMLLR with CMN used in the DNN-sMBR, is used. Another key feature in training this model is that rather than using vanilla SGD, a preconditioned SGD described in [19] is used in training. Details regarding the model and its training is provided in [22]. In summary, we train the neural net using pnorm where p=2 with input dimension of 2400 and output dimension of 300. The network consists of 4 hidden layer and is trained for 8 epochs where the learning rate decreases from 0.02 to 0.004. Then for the extra 4 epochs the learning rate remains constant.

3) DNN-TDNN-LSTM

Unlike previous models that uses a feed-forward network which cannot capture temporal difference between the frames, this model is composed of recurrent units that can model long term temporal dependency. Speech being a sequential input signal, sequential models have shown to outperform feed-forward networks in many speech recognition tasks.

Data for training this model is obtained by changing the speed of original data by a factor of 0.9, 1 and 1.1 along with the volume. LDA-MLLT and diagonal Universal Background Model (UBM) is applied to the MFCC features of the perturbed data. We then train ivector extractor, modify speaker info, extract ivectors and then align with fMLLR-lats. fMLLR transformation is applied as in [23]. We then train the neural net with i-vector input dimension of 100, output dimension of 3352. The network consists of 7 TDNN layers and 3 LSTM layers trained for 4 epochs with initial effective learning rate of 0.001 and final effective learning rate of 0.0001. As we are using a lattice-free model, it reduces the size of HCLG graph which helps in faster decoding.

VI. Results and Discussion

A. Baseline Model

Table III shows word error rates obtained for different combinations of speaker adaptation and feature vector optimization. Finally a combination of SAT, LDA and Delta-Delta with 5500 tri-phone states and 90000 gaussians was used as a baseline GMM-HMM model for training DNN model. It was observed that using features transformed with LDA and MLLT brought about around 15% relative improvement over the model trained with delta-delta features. Further application of speaker adaptive training produced a 12.2% additional relative improvement. So finally, the model trained using features transformed with LDA and MLLT trained with speaker adaptive training was selected as the baseline model.

B. DNN Models

Table IV shows the performance of the three models trained on 200 hours speech corpus. The speech corpus consisted of the 150 hours of open-sourced data coupled with 50 hours of speech corpus collected at Paaila. Alignments obtained from the application of the baseline model SAT-LDA-MLLT on the training data was used to train the three models. While DNN with pnorm non-linearity activations, DNN-Pnorm, resulted in a lighter model with faster decoding, it performed the worst of all. With RBM pretraining and using constrained MLLR features, the feed forward network performed very close to the LSTM based model. Finally, it was observed that recurrent neural networks outperformed the feed forward networks for the large vocabulary task. This is suggestive of the temporal dependence of sequence of phones which could be traced back to their origin where a meaningful utterance has a sequence of words which in themselves have rich temporal dependence.

TABLE III. WER for Different GMM-HMM Models

Training Method	**Data(hours)**	**%WER**	**%SER**	**Ins**	**Del**	**Sub**	total
Delta-Delta	80	40.27	61.24	3897	5981	2652	36330/90208
LDA-MLLT	80	36.14	55.8	5309	3959	23337	32605/90208
SAT-LDA-MLLT	80	31.351	49.70	5809	2702	19916	28427/90208
Delta-Delta	150	39.15	59.85	3880	5841	2559	35320/90208
LDA-MLLT	150	33.56	52.79	5091	3585	21595	30271/90208
SAT-LDA-MLLT	150	29.45	46.95	2656	2439	18441	26565/90208

TABLE IV. WER on Different DNN Models

Training Method	**Data(hours)**	**%WER**	**%SER**	**Ins**	**Del**	**Sub**	total
DNN-sMBR	200	12.30	22.27	2671	956	8542	12169/98929
DNN-Pnorm	200	18.16	29.14	4618	1075	12273	17966/98929
DNN-TDNN-LSTM	200	11.55	21.53	1610	1255	8563	11428/98929

VII. Conclusion

We have presented experiments with GMM-HMM and DNN-HMM systems with different training methodologies and optimization techniques on a 200 hours read speech corpus. Best performing GMM-HMM model - model trained with speaker adaptive training on features transformed with LDA and MLLT - was used as a baseline for the DNN-HMM model. DNN model using TDNN-LSTM units performed better than merely feed-forward networks. But performance of DNN with RBM-pretraining was comparable to the LSTM based model. Accuracy obtained for these models might alter with different test set which are very different from the training set used.

Future work on Nepali speech recognition should focus on incorporating spoken speech as it is very different from read Nepali speech. Also, use of end-to-end approaches may be explored in the future. Due to high phone to letter correspondence in Nepali, there is a good chance of much better performance with end-to-end models that can predict sequence of letters directly from speech features or raw speech data.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

All of the authors contributed to this research. The authors' contributions are as follows: Experiments, Elina Baral; Design and Methodology, Elina Baral and Sagar Shrestha; Writing-Original Draft, Elina Baral and Sagar Shrestha; Writing-Review and Editing, Sagar Shrestha and Elina Baral; Resources Sagar Shrestha and Elina Baral; Comparison and Analysis, Elina Baral and Sagar Shrestha.

Acknowledgment

The authors wish to thank Paaila Technology Private Limited for sponsoring this research.

References

[1] G. Hinton, *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.

[2] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1-3, pp. 133-147, 1998.

[3] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996, vol. 1, pp. 346-348.

[4] S. Molau, "Normalization in the acoustic feature space for improved speech recognition," Ph.D. dissertation, Bibliothek der RWTH Aachen, 2003.

[5] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171-185, 1995.

[6] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291-298, 1994.

[7] K. Shinoda, "Speaker adaptation techniques for automatic speech recognition," in *Proc. APSIPA ASC*, 2011.

[8] Kaldi toolkit. [Online]. Available: https://kaldi-asr.org/

[9] Summer Institute of Linguistics, *Nepali Segmental Phonology*, Kirtipur: Summer Institute of Linguistics, 1971.

[10] Nepali monolingual written corpus. [Online]. Available: http://catalog.elra.info/en-us/repository/browse/ELRA-W0076/

[11] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69-88, 2002.

[12] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. Fourth International Conference on Spoken Language Processing*, 1996, vol. 2, pp. 1137-1140.

[13] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75-98, 1998.

[14] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788-798, 2010.

[15] L. R. Bahl, P. F. Brown, P. V. D. Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. ICASSP*, 1986, vol. 86, pp. 49-52.

[16] Open sourced nepali speech corpus. [Online]. Available: https://www.openslr.org/54/

[17] Paaila technology. [Online]. Available: https://www.paailatechnology.com

[18] Training recipe for DNN-pnorm setup. [Online]. Available: https://kaldi-asr.org/doc/dnn1.html

[19] Training recipe for DNN-sMBR setup. [Online]. Available: https://kaldi-asr.org/doc/dnn2.html

[20] Training recipe for DNN-TDNN-LSTM setup. [Online]. Available: https://kaldi-asr.org/doc/dnn3.html

[21] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 599-619.

[22] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," arXiv preprintar Xiv:1410.7455, 2014.

[23] T. S. Nguyen, K. Kilgour, M. Sperber, and A. Waibel, "Improved speaker adaptation by combining i-vector and fMLLR with deep bottleneck net-works," in *Proc. International Conference on Speech and Computer*, 2017, pp. 417-426.

Elina Baral received her B.E. degree in Computer Science from Kathmandu University, Banepa, Nepal, in 2019. This research was produced during her internship at Paaila Technology Pvt. Ltd. She is currently a research intern at Nepal Applied Mathematics and Informatics Institute (NAAMII). Her current research interests include Natural Language Processing (NLP) and deep learning.

Sagar Shrestha received his B.E degree in Electronics and Communication Engineering from Pulchowk Campus of Tribhuvan University, Pulchowk, Nepal. He is currently with Paaila Technology as a co-founder and AI engineer since past three years. He has also worked as an assistant lecturer of Telecommunication and Instrumentation Systems at Thapathali Campus of Tribhuvan University, Nepal. His current research interests include signal processing, machine learning and robotics.