A Novel Deep Learning Based Nepali Speech Recognition

Basanta Joshi, Bharat Bhatta, Sanjeeb Prasad Panday(✉),
and Ram Krishna Maharjan

Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Nepal
{basanta,sanjeeb,mrrk}@ioe.edu.np

Abstract. Automatic speech recognition has allowed human beings to use their voices to speak with a computer interface. Nepali speech recognition involves conversion of Nepali language to corresponding text in Devanagari lipi. This work proposes a novel approach for developing Nepali Speech recognition model based using CNN-GRU. The data is collected from the Librispeech. The collected data is pre-processed and MFCC is applied on it for feature extraction. CNN-GRU model is responsible for extraction of the features and development of the acoustic model. CTC is responsible for decoding. The performance of the developed model has been assessed using Word Error Rate of the transcribed text.

Keywords: Automatic speech recognition · Nepali speech recognition · Recurrent Neural Network (RNN) · Convolution Neural Network (CNN)

1 Introduction

Speech recognition is the process of conversion of raw audio speech to its textual representation. Nowadays, a machine is able to recognize speech and convert the speech in the form of text. Information technology is an integral part of modern society. Recent advancement in modern communication devices helps to make life easier. Although the majority of the people live in the rural areas. Most of them are not able to read and write well. Speech recognition helps them to be familiar with the technology. To build an automatic speech recognition (ASR) system the prerequisite element is data-set. One of the reasons for not conducting the research work in Nepali ASR is due to insufficient Nepali spoken corpus. Currently available datasets are not enough to build a good Nepali ASR system. ASR systems can be developed by using the traditional or Deep Learning based approach.

In traditional approach, Speech Recognition requires separate blocks of phonetic/linguistic constructs, acoustic model and the language model and may involve hidden markov models (HMMs), gaussian mixture models (GMMs), hybrid HMM/GMM systems, dynamic time wrapping (DTW's), etc. But, in deep

S. Mekhilef et al. (Eds.): ICEEE 2022, LNEE 894, pp. 433–443, 2022.
https://doi.org/10.1007/978-981-19-1677-9_39

learning approach, the performance of the ASR system is significantly improved with using separate phonetic/linguistic constructs [1,2].

Most of the previous works that have been built now could not predict well for the unseen data. This is due to the lack of sufficient Nepali dataset (Nepali spoken corpus). Compared to previous models, the model also needs new architecture that must be able to predict well. And also to increase the performance of this ASR system, the model must be trained with a large amount of speech corpus.

This paper proposes a Nepali ASR system based on Deep Learning Approach. The system utilizes the MFCC for feature extraction, CNN to enhance the feature and GRU to create the acoustic model.

2 Review

There have been few research works carried out regarding development of Nepali speech recognition systems. One of traditional methods for Nepali Language based on HMM (Hidden Markov Model) is used for speaker independent isolated word Automatic Speech Recognition (ASR) system [3]. This system is trained and tested by collecting Nepali words from 8 different speakers in a room environment out of which 4 were male and 4 were female. The test cases consist of trained and untrained data. The recorded word was trained 20 times & tested 10 times and the overall accuracy of this ASR system is found to be 75% [3].

One of the first deep learning based approaches for Nepali Speech Recognition model uses RNN (Recurrent Neural Networks) for processing sequential audio data and CTC for maximizing the occurrence probability of the desired labels from RNN output [4]. On implementing a trained model, audio features are processed by RNN and Softmax layer successively. The output from the Softmax layer is the occurrence probabilities of different characters at different time steps. The task of decoding is to find a label with maximum occurrence probability. However, the model fails to accurately predict sounds like 'दु', 'वि', 'के', 'लि' treating them as single sounds.

These systems need some modification to increase the performance of the model. The size of the neural network and dataset must be significantly increased so that the network can learn accurately. The generalization capabilities of the model can be increased by adding a CNN layer. Some preliminary research regarding LSTM-CTC for Nepali Speech Recognition [5] has already been started by the same research group which is the baseline for this work.

3 Method

Feature extraction and model building (Front end processing) and decoding using lexicon and language model (back end) are the integral parts of the speech recognition system. Features are the attributes that provide uniqueness to objects i.e. they can be easily distinguished from others. An acoustic model contains statistical representations of each of the distinct sounds that makes up a word. The architecture of the model is described in Fig. 1.

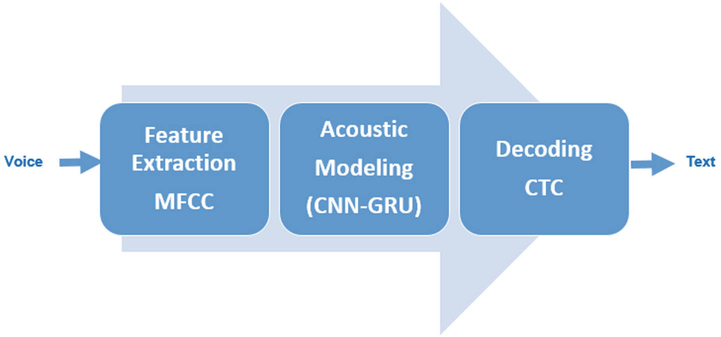


Fig. 1. Proposed ASR system

3.1 Feature Extraction

Features are the individual measurable property or characteristic of a phenomenon being observed, that can be used as a distinctive attribute or aspect of something. MFCC features are sequences of Acoustic feature vectors where each vector represents information in a small time window of signal [6]. The input analog speech after sampling at 16 kHz is represented as $S = [s_1, s_2, s_3, ..., s_n]$. Pre-emphasis of the input signal is done by using:

$$S^d[n_d] = S[n_d] - \alpha S[n_d - 1], where\, 0.95 < \alpha < 0.99 \quad (1)$$

Emphasis is carried out by taking $\alpha = 0.97$.

Speech signals are not stationary in nature. The main role of a window is to make signals stationary. The signal $S^d[t_d]$ is windowed to 25 ms by using a hamming window [7]. The consecutive interval between the two windows is 10 ms. If S[n] is sliced window, w[n] is window used on signal $S^d[t_d]$ then Sl[n] is given by Eq. 4.2.

$$Sl[n] = w[n]S^d[t_d] \quad (2)$$

where w[n] is hamming windows [7] given by relation 4.3

$$w[n] = 0.54{-}0.46cos\left(\frac{2\pi n}{N-1}\right) \quad (3)$$

After performing the windowing operation 512 point FFT has been taken. The output frequencies are modeled to human auditory systems. These FFT are passed into Mel- bank filters. The mel-frequency conversion is given by

$$M(f) = 1127 \ln\left(1 + \frac{f}{700}\right) \tag{4}$$

Power spectrum is represented by Mel filterbank. Since humans are less sensitive to small energy changes at high energy than small changes at a low energy level, logarithmic is taken from the output from the Mel filterbank. DCT is taken from the result after applying log energy computation. An orthogonal transformation is carried out by using the DCT. The DCT transformation produces uncorrelated features. The final MFCC-13 coefficients are represented by the $X = [x_1, x_2, x_3, , , x_T]$. These features need to normalize before training the model. MFCC of hundreds random samples are taken to compute mean and standard deviation. Then all the training features get normalized using the relation

$$X_n = \frac{X - \bar{X}}{\sigma + \Delta e} \tag{5}$$

Δe is a very small number to prevent division from zero.

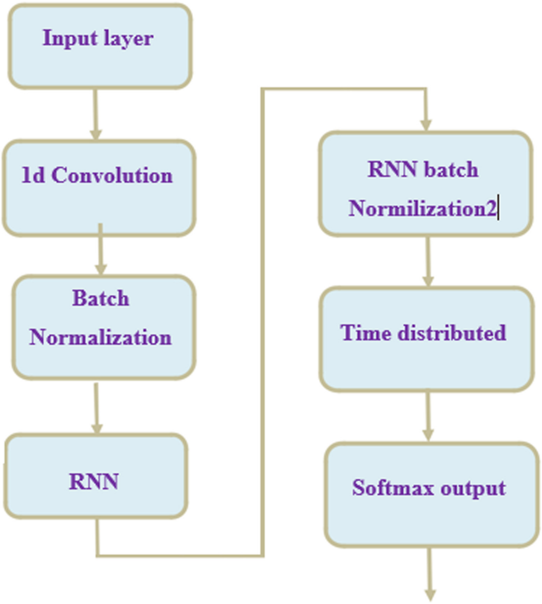


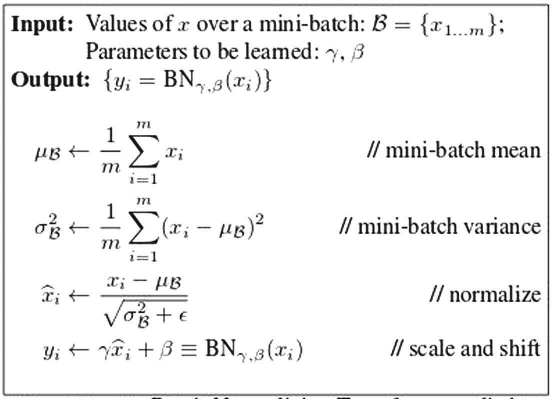
Fig. 2. Steps for Acoustic modeling

3.2 Acoustic Modeling and Decoding

A model or hypothesis is needed to build from the seen data in order to predict the unseen data. The model building is carried out by the neural network. The combination of CNN-GRU network is used to build the model. Here the CNN is used to summarize and collect useful features from the input and RNN in association with CTC is used to build the model. i.e. Deep learning model is used to build the model. Here the objective is to build the model and provide the posterior probability at the output so that the symbol having the greatest posterior probability is provided as output. The whole process can be summarized as shown in Fig. 2.

The input layer consists of 13- units. Thirteen features supplied by the MFCC are taken as input to these layers. CNN performs 1d-convolution. Convolution and max pooling are two major tasks of CNN responsible for generating the feature vector.

Batch normalization is used as per algorithm shown in Fig. 3 for normalization the output of CNN and then fed to RNN network.



```
Input: Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
       Parameters to be learned: $\gamma$, $\beta$
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

    $\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$                        // mini-batch mean

    $\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$          // mini-batch variance

    $\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$                        // normalize

    $y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$         // scale and shift
```

Fig. 3. Algorithm for Batch Normalizing Transform applied to activation x over a mini batch

Now, GRU is used to prepare the model for RNN. The output of GRU is again normalized and transferred to time-distributed layers and output is predicted by the Softmax function which involves calculating the posterior probability of each symbol. The decoding is carried out by the CTC [2]. The input to CTC is supplied from the output of the softmax. The decoded output of CTC is later mapped back to Nepali language characters. This mapped output is the predicted output of the model. The loss is computed by comparing the true labels and predicted labels. Later the model weights get adjusted.

3.3 Evaluation

Word Error Rate (WER) is the common evaluation metric for speech recognition system [8]. It is an average of the sum of substituted words, inserted words and deleted words of predicted output to total number of words in a ground sentence. The value of WER lies within 0 and 1 or 0% and 100%.

4 Experiments

4.1 Dataset

In this work, the dataset is the audio voice file which is provided by Open Speech and Language. Resources [9]. The dataset consists of high quality spoken corpus recorded by eighteen unique female speakers. Every voice clip is mapped with the respective transcriptions. The dataset is divided into parts: train and test set. The ratio of train and test is 80%:20%.

4.2 Experimental Setup

The training is carried out in a local machine with CPU Intel Core i7-8550U and GPU MX150 using python and its library. Table 1 and 2 summarizes the parameters used for Feature extraction using MFCC and CNN Network. In the case of GRU, 200 units with learning rate = 0.015, 100 epochs and momentum of 0.9 is taken during the training to obtain the best result.

Table 1. MFCC feature summarization

Parameter	Value	Description
Window	Hanning	Use for FIR Filter
FFT	256 point	Extract Frequency components
Window duration	20 ms	Size of each chunk
Max_frequecncy	8000 khz	Maximum Allowable Frequency
MFCC dimension	13	No of Features provided by MFCC
Mini-batch size	50	Size of each mini batch
Maximum duration of audio clip.	10 s	Audio data having length more than 10 s are not allowed during the training

4.3 Result and Analysis

Different experiments were carried out in CPU and GPU by varying parameters like learning rate, mini batch size, momentum, number of epochs. The summary of experiments performed is listed in Table 3.

Table 2. CNN parameters summarization

Parameters	Value	Description
Input dimension	13	Number of neurons in input layers
Filter	200	The dimensionality of the output space (i.e. the number of output filters in the convolution)
Conv_stride	2	Specifying the stride length of the convolution.
Dilation_rate	1	Specifying the dilation rate to use for dilated convolution
Activation function	ReLU	Specifying the activation function

Table 3. Summary of experiments performed

Parameters	Experiment1	Experiment2	Experiment3	Experiment4
Training data (Utterances)	16412	16412	1945	1945 (corpus)
Testing data (Utterances)	9588	9588	570	570 (corpus)
Learning rate	0.03	0.03	0.05	0.015
Momentum	0.9	0.9	0.9	0.9
Batch size	100	300	20	50
Epochs	400	100	100	100
Training machine	CPU Intel Core i7-8550U	CPU Intel Core i7-8550U	CPU Intel Core i7-8550U	GPU NVIDIA GEFORCE MAX 150
Training duration	1.5 days	1.3 days	1.3 days	1.5 h

The first experiment is carried out as per parameters and configuration mentioned in Table 4 and training Loss and validation loss curve is shown in Fig. 4. From the graph it can be concluded that the training loss is much less than validation loss. Hence the system gets over fitted. The system can perfectly classify the seen data (data that involves training) but doesn't classify the unseen data.

The second experiment is carried out as per parameters and configuration mentioned in Table 5 and training Loss and validation loss curve is shown in Fig. 5. The third experiment is carried out as per parameters and configuration mentioned in Table 6 and training Loss and validation loss curve is shown in Fig. 6. During the training after 45 epochs, sudden abrupt discontinuity is observed. The training strikes in local maxima. And with an increase in the number of iterations the model does not learn any things. The training loss and validation loss remain unchanged. To eliminate this problem we can apply two

Table 4. Summary of first experimental setup

Model	CNN+BN+GRU+BN+Dense layer+softmax+CTC
GRU units	200
Filter size	200
Kernel size	11
MFCC	dimension = 13, window = 25 ms, filter-bank = 26
Mini-batch size	100
Optimizer	SGD (learning rate = 0.03, decay = 1e−6, momentum = 0.9
Epoch	400

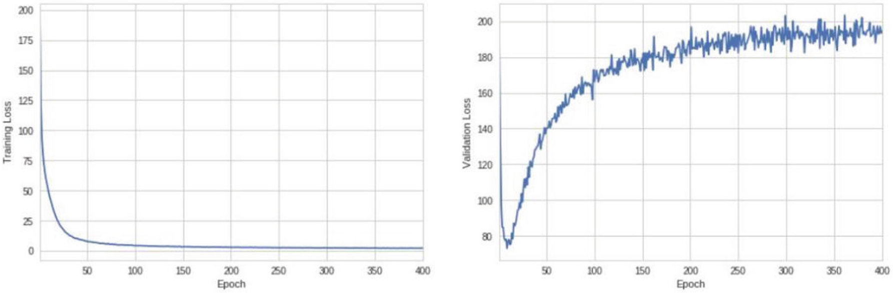


Fig. 4. Training loss and validation loss curve of first experiment

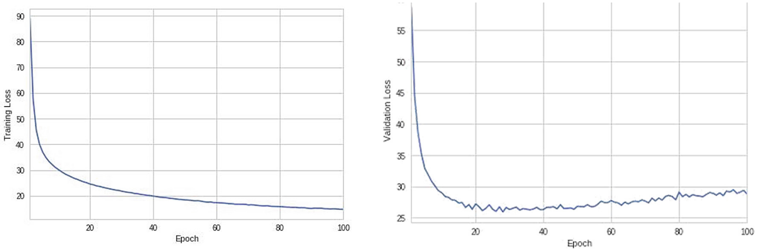


Fig. 5. Training loss and validation loss curve of second experiment

techniques. First is to rearrange the network and other to increase the value of dropout. Re-arranging the network may be a tedious task so first the dropout is increased to 0.7 from 0.5. No prediction is made and another experiment is set up. The fourth experiment is carried out as per parameters and configuration mentioned in Table 7 and training Loss and validation loss curve is shown in Fig. 7.

Some sample of the output of the model are listed as Table 8.

The summary of the different experiments can be visualized as shown in Fig. 8.

Table 5. Summary of second experimental setup

Model	CNN+BN+GRU+BN+Dense layer+softmax+CTC
GRU Units	200
Filter size	200
Kernel size	11
MFCC	dimension = 13, window = 25 ms, filter-bank = 26
Mini-batch size	300
Optimizer	SGD (learning rate = 0.03, decay = 1e−6, momentum = 0.9
Epoch	100

Table 6. Summary of third experimental setup

Model	CNN+BN+GRU+BN+Dense layer+softmax+CTC
GRU Units	200
Filter size	200
Kernel size	11
MFCC	dimension = 13, window = 25 ms, filter-bank=26
Mini-batch size	20
Optimizer	SGD (learning rate = 0.05, decay = 1e−6, momentum = 0.9
Epoch	100

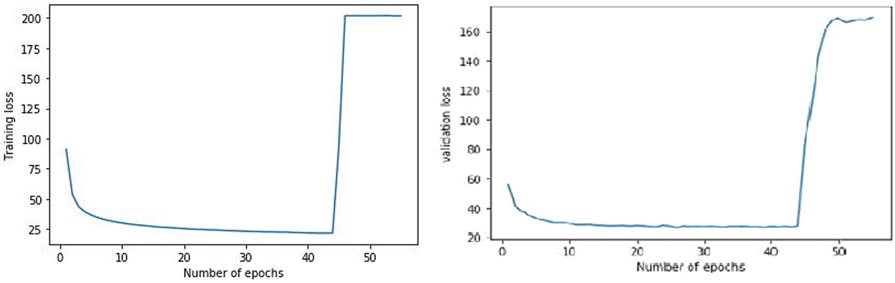


Fig. 6. Training loss and validation loss curve of third experiment

Table 7. Summary of forth experimental setup

Model	CNN+BN+GRU+BN+Dense layer+softmax+CTC
GRU Units	200
Filter size	200
Kernel size	11
MFCC	dimension = 13, window = 25 ms, filter-bank = 26
Mini-batch size	50
Optimizer	SGD (learning rate = 0.015, decay = 1e−6, momentum = 0.9
Epoch	100

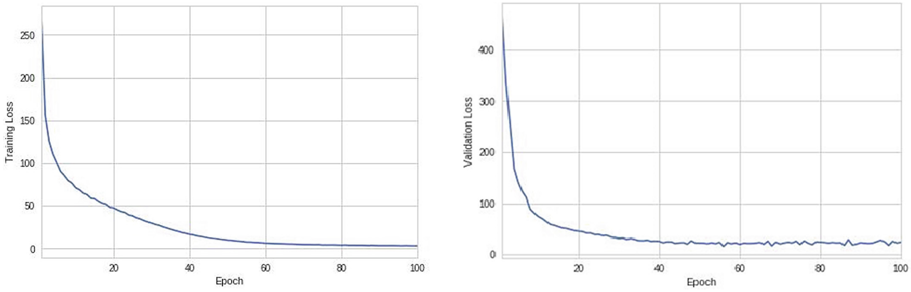


Fig. 7. Training loss and validation loss curve of fourth experiment

Table 8. Sample Results of Nepali ASR

Ground Truth	Model prediction	WER
इन्डोनेसियाली पपुवा प्रान्तमा रहेको राष्ट्रिय निकुञ्ज	इन्डोनेसियाली पपुवा प्रान्तमा रहेको राष्ट्रि निकुञ्ज	0.1667
चलचित्रमा केमियो रोलमा नायक राजबल्लभ कोइरालालाई पनि देख्न पाइनेछ	चलचित्रमा केमियो रोलमा नायक राजबल्लभ कोइराँलालाई पनि देख्न पाइनेछ	0.111
उनले दुई हजार दसमा जर्जियामा सुरु हुने स्टर्स टुर्नामेन्टमा भाग लिन थाले	उनले दुई हजार दसमा जर्जियामा सुरु हुने स्टोर्स टुर्नामेन्टमा भागलिन थाले	0.250

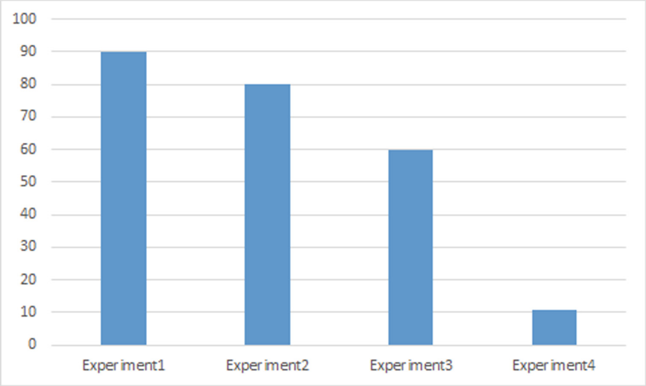


Fig. 8. Training Loss and Validation loss curve of fourth experiment

5 Conclusion

In this work, an approach for nepali speech recognition CNN-GRU has been proposed. The MFCC is used for feature extraction and takes window size of 20 ms, FFT of 256 points and hanning windows as FIR filter. The dimension of MFCC is 13. These MFCC features are applied to CNN. Where CNN performs 1-d convolution and pooling to extract useful features. The features are applied to GRU network which is used for acoustic model building. After hyperparameter tuning, the best model is built by taking the parameters learning rate 0.015, batch size 50, momentum 0.9 and dropout rate of 0.65. The built system effectively converts correctly spoken Speech to Nepali text. The WER is found to be 10.67%. The system performance can be increased by adding the Lexicon and building Language model, which can be done in future. Training the system with a large number of valid corpus will increase the system performance.

Acknowledgement. This work has been supported by the University Grants Commission, Nepal under a Faculty Research Grant (UGC Award No. FRG-76/77-Engg-1) for the research project "Preparation of Nepali Speech Corpus: Step towards Efficient Nepali Speech Processing".

References

1. Karita, S., et al.: A comparative study on transformer vs rnn in speech applications. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 449–456. IEEE (2019)
2. Passricha, V., Aggarwal, R.K.: Convolutional neural networks for raw speech recognition. In: From Natural to Artificial Intelligence-Algorithms and Applications. IntechOpen (2018)
3. Ssarma, M.K., Gajurel, A., Pokhrel, A., Joshi, B.: Hmm based isolated word nepali speech recognition. In: 2017 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 71–76. IEEE (2017)
4. Regmi, P., Dahal, A., Joshi, B.: Nepali speech recognition using rnn-ctc model. Int. J. Comput. Appl. **178**(31), 1–6 (2019)
5. Bhatta, B., Joshi, B., Maharjhan, R.K.: Nepali speech recognition using CNN, GRU and CTC. In: Proceedings of the 32nd Conference on Computational Linguistics and Speech Processing (ROCLING 2020), pp. 238–246 (2020)
6. Gupta, R., Sivakumar, G.: Speech recognition for Hindi language. IIT BOMBAY (2006)
7. Kopparapu, S.K., Laxminarayana, M.: Choice of mel filter bank in computing MFCC of a resampled speech. In: 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), pp. 121–124. IEEE (2010)
8. Kong, X., Choi, J.Y., Shattuck-Hufnagel, S.: Evaluating automatic speech recognition systems in comparison with human perception results using distinctive feature measures. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5810–5814. IEEE (2017)
9. Sodimana, K., et al.: A step-by-step process for building TTS voices using open source data and framework for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese. In: Proceedings of the 6th International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU), pp. 66–70, Gurugram, India, August 2018