Indian Journal of Science and Technology, Vol 10(40), DOI: 10.17485/ijst/2017/v10i40/120273, October 2017
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Urdu Language Translator using Deep Neural Network

Syed Abbas Ali[1]*, Sallar Khan[2], Humaira Perveen[1], Reham Muzzamil[1], Mahnoor Malik[1] and Faiza Khalid[1]

[1]NED University of Engineering and Engineering and Technology, Main University Road، Karachi 75270, Pakistan; Saaj.research@gmail.com, humaira.0031@yahoo.com, reham.muzammil@yahoo.com, mahnoormalikcs54204@yahoo.com, fzk_1995@yahoo.com
[2]Sir Syed University of Engineering and Engineering and Technology, Karachi, Pakistan; SallarKhan_92@yahoo.com

Abstract

Urdu language of Pakistan has more than 100 million speakers in Pakistan, India, Afghanistan and Middle East. With low English literacy rate average Urdu speaking person faces barriers in communicating with foreign people in terms of accessing information, carrying business. This paper proposes an interactive Urdu to English language speech translator using deep Neural Network. ASR module in proposed pipeline is composed of deep neural network and is simpler as compared to traditional ASR which requires complex hand engineering like feature extraction and resources like phoneme dictionary. It was clearly seen that the proposed model shows the high accuracy when the input is recorded audio and it shows poor performance with real time input. While one HTTP request per input transcription produced English translation for Text to Text translation using Python Text Blob library. The final output was achieved with a delay of no more than 30 seconds. Furthermore, we have tested and provided some statistical findings, the result shows that value updating for neural network layer's bias, standard deviation when Adam optimizer parameters are set as follows: beta1=0.9, beta2=0.9 and learning rate =0.01 meanwhile dropout rate was kept to 5% to offer regularization and observed value for scalar maximum lies between 0 and 0.08. There is a little deviation at 0.05 step, value decreases and afterwards that bias maximum scalar increases with positive values and finally increases exponentially at later stages of training further results are discussed in experiment section respectively. The proposed speech recognition model out performs traditional automatic speech recognition systems in efficiency, simplicity and robustness.

Keywords: Deep RNN, Language Translator, N-gram LM, Text Blob Translation, Urdu Language, ASR

1. Introduction

The model for Urdu to English speech translator consists of three modules namely Automatic Speech Recognition, Text to Text translation and Text to Speech conversion. All three modules are built on techniques to give effective performance.

Traditional ASRs employ complex techniques based on Hidden Markov Models/ Gaussian Mixture Models where HMMs normalize temporal difference and GMMs compute emission probabilities of HMM states. Recently HMM/DNNs have outperformed GMM in terms of speech frames classifications into context dependent clusters (senones). This introduction of Deep Learning Algorithms in Traditional ASR usually improves only acoustic models of traditional ASR thus deep learning still plays a minor role in traditional speech recognition pipeline.

To further reduce the complexity of ASR and to improve its accuracy researchers have proposed approach of end to end ASR that is direct mapping of speech and labels using neural network without any intermediate components. By carrying out this end to end approach for Urdu speech recognition we simplify the traditional ASR pipeline composed of multiple training stages, various resources like dictionaries, decision trees, hand

*Author for correspondence

engineered input features and acoustic models. With this approach we are able to maximize the efficiency and effectiveness of applying Deep Learning.

Implementing end to end deep learning poses some challenges like (i) availability of large training data. (ii) effective utilization of this large data. The requirement of labeled training data set was fulfilled by acquiring Urdu language corpus from CLE (Centre of Language engineering Lahore).

Training of large dataset requires alignment of input speech with labeled training data. Speech recognition is a sequence problem that requires prediction of label sequence from input acoustic signal. The problem was addressed by solution proposed by Gomes, Fernandez, Schnidhuber and Graves using Connectionist Temporal Classification[1].

The second module translates Urdu transcription to English text. Considering real time translation, we require accurate translations for very large range of Urdu Language vocabulary which made us opt for the use of effective online natural language processing API of Python – TextBlob.

Third module of English Text to English Speech is implemented using IBM Watson API of text to speech conversion. Offline models for text to text or text to speech conversion often are suitable for limited vocabulary. Use of online APIs assisted translation of wide Urdu Language vocabulary to English Text and speech in almost real time.

In section 2, the related literature review is covered. In section 3 Methodology adopted is described. Section 4 presents training and experiment conducted. Section 5 is brief overview of related work. Conclusion is proposed in section 6.

2. Literature Review

This section covers speech recognition techniques with a focus on Urdu language. Use of State of the art technique Deep learning in ASRs is also discussed. A brief overview of language models and techniques to employ them for language translation is discussed.

2.1 Urdu Speech Recognition

Continuous Urdu speech recognition over large vocabulary is described in[2] where CMU Sphinx Tool Kit is used to develop acoustic and language models for Urdu Speech. This system is developed specifically for Lahore suburban accent.

[3]Addresses major hurdle of unavailability of large amount of transcribed spontaneous speech data for ASR development. This research proposes a single speaker, medium vocabulary speech recognition system for Urdu by training a mixture of read and spontaneous speech data. A statistical based approach is undertaken in[4] for Urdu ASR. HMM model is used to develop ASR for speaker independent, small size vocabulary of 52 isolated most commonly spoken Urdu words. The proposed system is developed on Sphinx4 framework. [5]Investigates 3 methods for Urdu ASR. Work includes extraction of Mel Frequency features and then training them on Support Vector Machine (SVM), Random Forest (RM) classifier, and a Linear Discriminate Analysis classifier (LDA). Comparison shows better performance on SVM for this particular task.

2.2 Deep learning and Speech Recognition

Baidu Deep Speech employs state of the art technique of Deep learning. Optimized RNN is used to model ASR independent of hand engineered components and robust to noisy background and speaker variation[6]. Deep Bi-directional LSTM is investigated in[7] for acoustic modeling. It outperforms GMM when applied on a part of Wall Street Journal dataset. Moderate improvement in WER is observed in[8]. Conducted same research of Deep Learning based acoustic modeling in speech recognition as compared to HMM and GMM. It adopts an alternative procedure of using feed forward neural network predicting posterior probabilities taking several input frames of coefficients. Context dependent-DNN-HMM are proposed for speech to text transcriptions in[9]. Approach integrates HMMs and Deep Belief Network (DBN) for pre training of over 9000 tied states using 9 hidden layers of neural network.

2.3 Deep learning and Speech Recognition

Advanced approach of combining statistical machine translation and transliteration is proposed in[10] probabilistic models of conditional probability and joint probability were proposed. This approach considered both translation and transliteration when translating Hindi words. Effective results indicate usefulness of transliteration in translation of text. This[11] paper describes phrase based machine translation using unsupervised character based

models for Urdu, English and other language pairs. Different approaches are under taken for different languages. Improved Machine Translation is presented in[12] by pre ordering of sentences. Most of such work includes parser in source language to map words from source language to target language. This research overcomes need of source language parser by directly re ordering source side sentences for mapping to target word order using parallel corpora. This work is performed on Urdu to English, Hindi to English and English to Hindi[13] used statistical and language independent approach for information retrieval using n-grams. This research is proposed to handle multi lingual collection of documents. N-gram technique is discussed for developing Information Retrieval (IR) and NLP based tools. N-gram based language modeling is done for Urdu, Hindi etc.

3. Methodology

Proposed model of Urdu to English speech translator consists of three module pipeline.

3.1 Automatic Speech Recognition (ASR)

Speech recognition is done using state of the art technique of deep learning. Recurrent Neural Network is trained on Urdu data set of 708 Sentences and maps input speech to Urdu transcriptions.

3.2 Text to Text Translation

This module of pipeline is built using natural language processing library of python-TextBlob which translates Urdu text to English text in real time.

3.3 Text to Speech Conversion

Text to Speech Conversion module uses IBM Watson API for text to speech conversion. For efficient and correct results from last two modules, online API was used to cover large vocabulary instead of offline models which offer limited support in terms of data size.

3.4 RNN Setup

The core of ASR module is bi directional recurrent neural network (RNN) trained to input audio spectrograms and produce Urdu text transcriptions.

The RNN is supposed to map an input audio x to the transcription y, where y is sequence of probabilistic characters for Urdu transcription. With $\hat{y}$= P(c(t) |x), where c(t) {ش, ص, ض, ط, ظ, ع, غ,ف,ق,ک,گ, ل,م, ن, ں, و,ھ, ء,ی,ے, ا ب پ ت ٹ ث ج چ ح, خ, د,ڈ, ذ,,ر ڑ, ز,ژ,س, blank}.

Suppose a training set X = {(x (1), y (1)), (x (2), y (2)), ...} where x(i) refers to a single audio utterance and y as a label. Each audio is of time length T(i) where each time frame is a vector of some audio features, $x_t(i)$, t = 1,..., T (I).

We build our RNN with 5 layers of hidden processing units. The hidden unit at layer l is given by h(l) and h(0) refers to the input.

We made first three layers as non-recurrent. Output of first layer is dependent on audio frame(spectrogram) xt. Other non-recurrent layers for each time frame are independent of data. On each timestamp the remaining non-recurrent layers operate on independent data. Thus, the first 3 layers for each time t, are computed by:

$$h_t(l) = g(W(l)\, h_t(l-1) + b(l))$$

where, g(z) = min{max{0, z}, 20} is the clipped rectified-linear (ReLu) activation function and W(l), b(l) are the weight matrix and bias parameters for layer l. The fourth layer is a bi-directional recurrent layer. This bi-directional recurrent layer consists of two sets of hidden units: first set of hidden units with forward recurrence, that is h(f), while the other set with backward recurrence that is h(b):

$$h_t(f) = g(W(4)\, h_t(3) + W_r(f)\, h_{t-1} + b(4))$$
$$h_t(b) = g(W(4)\, h_t(3) + W_r(b)\, h_{t+1} + b(4))$$

Note that $h_t(f)$ must be computed sequentially from t = 1 to t = T (i) for the i'th utterance, while the units $h_t(b)$ must be computed sequentially computation from T=1 to T(I) gives h(f) and reverse computation from t=T(I) to 1 computes hb units.

$$h_t(5) = g(W(5)h_t(4) + b(5)) \text{ where } h_t = h_t(f) + h_t(b).$$

The output layer is a standard softmax function that yields the predicted character probabilities for each time slice t and character k in Urdu alphabets(haroof):

$$h_{t,k}(6) = \hat{y}_{b,k} \equiv P(c_t = k|x) = \frac{\exp(W_k(6)h_t(5) + b_k(6))}{\sum_j \exp(W_j(6)h_t(5) + b_j(6))}$$

b_k denote the k'th column of the weight matrix and k'th bias, respectively.

After predicting $P(c_t|x)$, next task is computation of CTC loss $L(\hat{y}, y)$ to measure error in prediction. The gradient $\hat{y}$ $L(\hat{y}, y)$ evaluation can be done during training by considering RNN outputs given the accurate (ground-truth) output, y, composed of character sequence. This gradient computation may be done using back propagation through the rest of the network; Adam optimizer is used for this purpose[14].

The Baidu Deep Speech RNN setup was referenced for speech recognition[15].

3.5 Language Model

RNN model trained from large data sets is capable of mapping input audio to labeled data and produce transcription based on characters, but error may occur on basis of phonetic plausible renderings based on words that may never or rarely occur in our training data. To reduce such errors, language model is used with the neural network. The language model is built from Urdu text corpus obtained from Center of Language Engineering (CLE) using open source toolkit of Kenlm.

3.6 KENLM Working

Given Urdu text corpus of CLE, the n-gram language model was trained using Kenlm. Kenlm employs smoothing technique to adjust n-grams to make better estimation of most probable Urdu sentences. Kenlm proved to be memory and time efficient in implementation. The trigram Urdu language model is built and then compiled into binary format for efficient loading time. Evaluation was done with different Urdu sentences with this language model and optimistic scores were obtained.

4. Training

The neural network is trained over Urdu language data set obtained from CLE Lahore. This phonetically rich corpus is based on 708 sentences which cover all 36 phonemes of Urdu language and total number of 5,656 words. This data set also contains recorded speech of these 708 Urdu sentences as both continuous speech and isolated words. The implementation of neural network was done on system with following specification: Processor: Intel® Core™ i5-2520M CPU @ 2.50GHz × 4 Disk: 92.6 Memory: 3.7 GiB Operating System: Ubuntu 16.04 LTS.

After building the language model using Kenlm and setting up the bi-directional recurrent neural network training is done as follows:

(i) Nine Urdu conversational sentences were selected and trained by the neural network. Checkpoint directories of training were stored using Tensor flow toolkit.

(ii) Testing of ASR was performed by using the checkpoint files.

We initiated our training with epoch number of 50 and gradually increased this number till neural network got fully trained on them to give results with complete accuracy. The following Table 1 shows the number of epochs at which the word error rate of a particular sentence became zero. Table 1 also shows observed results of training samples.

Table 1. Training samples

Urdu Sentences	Epochs required for RNN training with 0.0 WER
1 ری خبص	139
2 ےھتوقیوھباساپےکپآیک	104
3 جیبلکل	54
4 ںیلئاسمسبقمیکچھمےاسس	54
5 ںیھےنھچوپسکرامےسپآےھجم	139
6 اگوھبکبٹیسٹیارامھ	109
7 اگےیئآیمیںآپبعدںیھن	94
8 ںوھفورصمیںوقتسا	99
9 ےھانھچوپھچکےسپآےھجم	139

5. Experimental Results

ASR module in proposed pipeline is composed of deep neural network and is simpler as compared to traditional ASR which requires complex hand engineering like feature extraction and resources like phoneme dictionary. The proposed system when trained on 12-13 hours of speech data on nearly 8 GPUs is capable of giving accurate results in real time while proving robust to noisy environment and speaker accent variations. Text to Speech module of proposed pipeline uses TextBlob, online Python API for Urdu to English text translation. Final module of Text to Speech conversion is implemented using IBM Watson

API. After training is done checkpoint directories are maintained so that they can be used for testing. Testing is done for two input types recorded and real time shown in Tables 2 and 3 respectively.

Table 2. Recorded input

WER	Loss	AVG_CER	Sentences
0.00	0.60	0.00	ریخ بح بص
0.00	1.35	0.00	تقو یھبا سا پے ک پ آا ک ے ھ
0.00	0.43	0.00	چ ی بل کل
0.00	0.33	0.00	ق بس ا س ے ھچ مم س ا ئل ہ ی ں ھچ ک ی م
0.00	2.70	0.00	آپ س ے ما ر ک س پ و چ ھ ن ے ھ ی ں م چ ھ ے
0.00	3.10	0.00	ھ ما را ٹ ی س ٹ ک ب ھو گ ا
0.00	0.85	0.00	ن ھ ی ں آپ ب ع د م ی ں آ ئ ی ے گ ا
0.00	2.14	0.00	اس وقت م ی ں م ص رو ف ھو ں
0.00	2.51	0.00	آپ س ے ک چ ھ پ و چ ھ ن ا ھ ے م چ ھ ے

Table 3. Real time input

WER	Loss	AVG_CER	Sentences
1.00	64.76	1.00	ص ب ح ب خ ی ر
1.40	61.53	0.72	ک ے پ اس ا ب ھ ی وقت ھے ک ی آ پ
2.00	17.05	0.55	چ ی ب ل ک ل
2.00	16.25	0.62	ی ں اس س ب ق م ی ں ے ھ چ م ک چ ھ م س ا ئل
1.00	158.76	0.82	س ے ما ر ک س پ و چ ھ ن ے ھ ی ں م چ ھ ے آپ
3.00	98.76	0.75	ھ ما را ٹ ی س ٹ ک ب ھو گ ا
1.00	114.90	0.63	آپ ب ع د م ی ں آ ئ ی ے گ ا ن ھ ی ں
1.75	122.5	0.73	اس وقت م ی ں م ص رو ف ھو ں
2.16	103.81	0.73	آپ س ے ک چ ھ پ و چ ھ ن ا ھ ے م چ ھ ے

Since the dataset was limited, this can be clearly seen from the Table 3 that the model shows high accuracy when the input is recorded audio and it shows poor performance with real time input.

Next module in pipeline for Text to Text translation read these Urdu transcriptions and with one HTTP request per input transcription produced English translation using Python Text Blob library.

The translated text is converted to speech using Text to Speech API. The final output was achieved with a delay of no more than 30 seconds.

Deep RNN based ASR is built using Tensorflow toolkit. Tensorboard was used to plot computations and visualize recurrent neural network layers for different parameter values.

In Figure 1, the graph shows value updating for neural network layer's bias, standard deviation when Adam optimizer parameters are set as follows:

beta1=0.9, beta2=0.9 and learning rate =0.01. Dropout rate was kept to 5% to offer regularization.

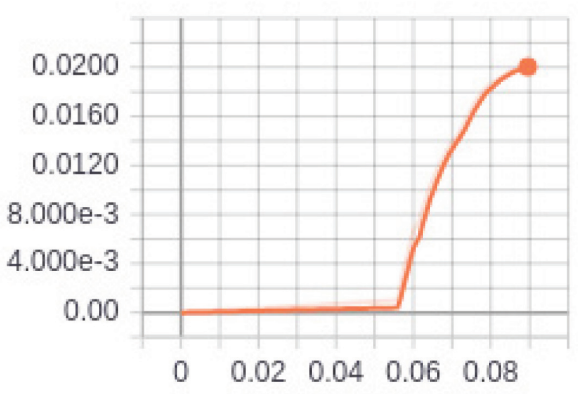

Figure 1. Bi-direction RNN layer/backward/bias/max.

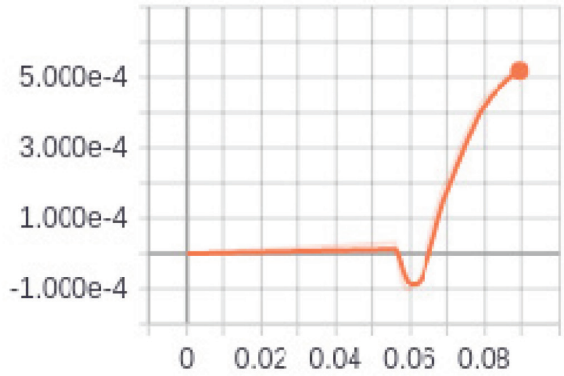

Figure 2. Bi-direction RNN layer/backward/bias/mean.

In Figure 2 The graph shows that mean scalar value of bias tensor variable for bi directional layer of RNN changes

the following values during training. Observed value for scalar maximum lies between 0 and 0.08. There is a little deviation at 0.05 step, value decreases and afterwards that bias maximum scalar increases with positive values and finally increases exponentially at later stages of training.

In Figure 3 The graph shows that standard deviation scalar value of bias tensor variable for bi directional layer of RNN changes the following values during training. Observed value for scalar maximum lies between 0 and 0.08 and shows that standard deviation scalar remains zero for earlier training epochs and increases exponentially at later stages of training over time.

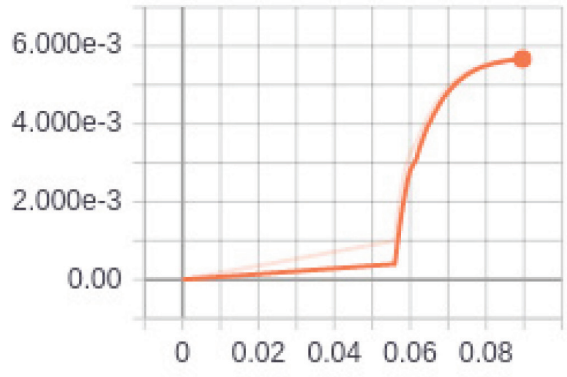

Figure 3. Bi-direction RNN layer/backward/bias/sttdev.

In Figure 4 The graph shows that minimum scalar value of bias tensor variable for bi directional layer of RNN changes the following values during training. Observed value for scalar maximum lies between 0 and negative 0.08 and shows that standard deviation scalar remains zero for earlier training epochs and at step of around 0.06 starts increasing negatively. Graph shows negative exponential curve for later training epochs that is around 49, 50.

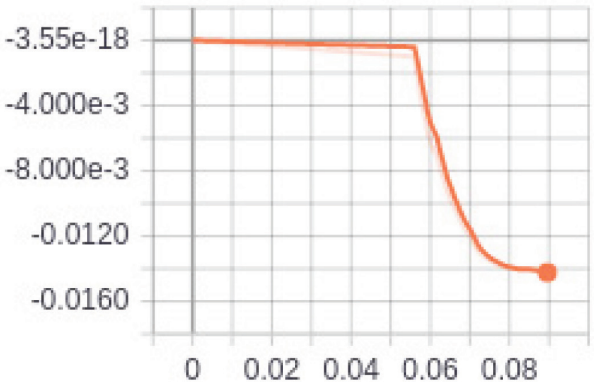

Figure 4. Bi-direction RNN layer/backward/bias/min.

In Figure 5 the graph shows histogram for bias values of forward recurrent layer of Bi directional RNN. Histogram is made of contour lines. Contour is 3 dimensional representations where x axis is showing range of values between -0.015 to 0.015. Contour at 49 step shows that total 494 samples (494 hidden units of our neural network) take values for bias in complete range of -.015 to 0.015. Spike of 18.3 means 18.3 hidden units have value of -0.0618.

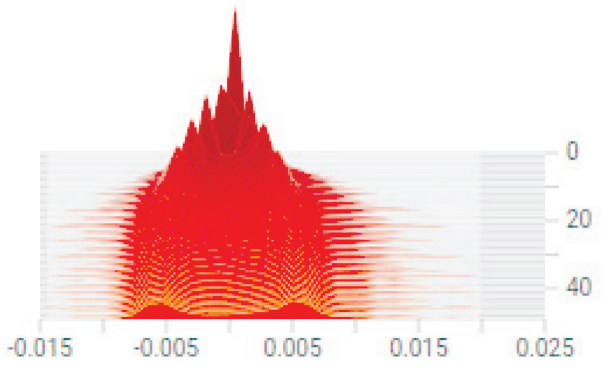

Figure 5. Bi-directional rnn layer/fwd/bias Histogram.

In Figure 6 The Histogram graph shows optimization gradient calculated over the period of time for tensor variable (bias) across forward layer of bi direction RNN. This shows that biases of all 494 hidden units of our neural network has gradient optimization of -0.0373 at training step of 49.

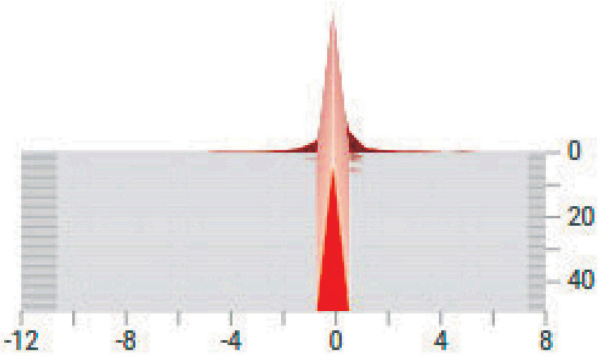

Figure 6. Bi-diectional RNN layer/fwd/bias/gradient Histogram.

The x axis shows the total range of gradients computed and spikes show the value taken by input units of neural network (in this case bias at bi-direction layer) at any training step.

The work is partially inspired by End to End Deep Speech system of speech recognition by Baidu. Several techniques employed are inspired by previous related work such as CTC loss function for error reduction in producing Urdu transcriptions. The work by Graves et al is on CTC loss function with LSTMs, we in contrast opted for simpler and effective Bi-RNN.

The proposed model in this paper is one of initial works carried out for real time speech translation of Urdu language to English using end to end Deep Learning technique for Urdu speech recognition. Microsoft Skype Translator and Google Translator API have recently moved to deep learning paradigm for speech translation including Urdu language as announced in August 2017. This research was carried out with Urdu Language Speech corpus obtained from Center of Language Engineering Lahore[16].

6. Conclusion

In this paper, an initial work carried out for an interactive speech translator from Urdu to English Languages with three module pipeline. ASR is based on end to end approach of Deep Learning and out performs traditional ASR in terms of simplicity and robustness to noise and speaker variation. Effective integration of online APIs for text to text translation and text to speech conversion achieve accurate results and in near real time. The proposed model established the proposition of achieving effective and accurate results in real time by increasing the computation power and size of language corpus. Once we scale the processing system and enhance the size of Urdu corpus, proposed model is capable to do real time translations from Urdu to English language.

7. References

1. Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist temporal classification. Proceedings of 23rd International Conference on Machine Learning - ICML '06; 2006. p. 369–76. Crossref.
2. Sarfraz H, Hussain S, Bukhari R. Large vocabulary continuous speech recognition for Urdu. Proceedings International Conferences. RANLP-2009; 2009. p. 246–50.
3. Raza A, Hussain S, Sarfraz H. An ASR System for Spontaneous Urdu Speech. Proceedings Orient; 2010. p. 1–6.
4. Mathias L. Statistical machine translation and automatic speech recognition under uncertainty. 2007 Dec.
5. Ali H, Jianwei A, Iqbal K. Automatic speech recognition of Urdu digits with optimal classification approach. International Journal of Computer Applications. 2015; 118(9):1–5. Crossref.
6. Amodei D, Anubhai R, Battenberg E, Case C, Casper J et al. Deep Speech 2: End-to-end speech recognition in English and Mandarin. 2015; 48.
7. Mohamed A, Seide F, Yu AD, Droppo J, Stolcke A et al. Deep bi-directional recurrent networks over spectral windows. 2015 IEEE IEEE Automatic Speech Recognition and Understanding, ASRU 2015; 2016; p. 78–83.
8. Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding; 2015. p. 167–74.
9. Dahl GE, Yu AD, Deng LL, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing. 2012; 20(1):30–42. Crossref.
10. Durrani N, Sajjad H, Fraser A, Schmid H. Hindi-to-Urdu machine translation through transliteration. Proceedings 48th Annual Meeting of the Association for Computational Linguistics; 2010 Jul. p. 465–74.
11. Denkowski M, Lavie A. Meteor universal: Language specific translation evaluation for any target language. Proceedings of the Ninth Workshop on Statistical Machine Translation; 2014. p. 376–80. PMCid:PMC4469275
12. Visweswariah K, Rajkumar R, Gandhe A, Ramanathan A, Navratil J. A word reordering model for improved machine translation. EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing; 2011. p. 486–96.
13. Majumder P, Mitra M, Chaudhuri BB. N-gram: A language independent approach to IR and NLP. Int. Conf. Univers. Knowl. Lang. 2002 Nov; 2.
14. Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY. On optimization methods for deep learning. Proceedings of 28th International Conference on Machine Learning (ICML 2011); 2011. p. 265–72.
15. Baidu Deep Speech RNN setup for speech recognition [Internet]. Available from: https://github.com/mozilla/DeepSpeech.
16. Urdu language Speech corpus from Center of Language Engineering Lahore [Internet]. Available from: http://www.cle.org.pk/clestore/index.htm.