Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 2, August 2022, pp. 1083~1090
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i2.pp1083-1090

Hindi to English transliteration using multilayer gated recurrent units

Mohd Zeeshan Ansari[1], Tanvir Ahmad[1], Mirza Mohd Sufyan Beg[2], Faiyaz Ahmad[1]
[1]Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi, India
[2]Department of Computer Engineering, Aligarh Muslim University, Aligarh, India

Article Info

Article history:

Received Sep 23, 2021
Revised May 25, 2022
Accepted Jun 10, 2022

Keywords:

Encoder decoder
Gated recurrent units
Sequence to sequence model
Transliteration

ABSTRACT

Transliteration is the task of translating text from source script to target script provided that the language of the text remains the same. In this work, we perform transliteration on less explored Devanagari to Roman Hindi transliteration and its back transliteration. The neural transliteration model in this work is based on a sequence-to-sequence neural network that is composed of two major components, an encoder that transforms source language words into a meaningful representation and the decoder that is responsible for decoding the target language words. We utilize gated recurrent units (GRU) to design the multilayer encoder and decoder network. Among the several models, the multilayer model shows the best performance in terms of coupon equivalent rate (CER) and word error rate (WER). The method generates quite satisfactory predictions in Hindi-English bilingual machine transliteration with WER of 64.8% and CER of 20.1% which is a significant improvement over existing methods.

Corresponding Author:

Faiyaz Ahmad
Department of Computer Engineering, Faculty of Engineering and Technology
Jamia Millia Islamia University
Maulana Mohammad Ali Jauhar Marg, New Delhi, 110025, India
Email: fahmad1@jmi.ac.in

1. INTRODUCTION

The transliteration from a source to a target script is defined as writing the text using the letters of the target language provided that the language of the text does not change. Moreover, it preserves the pronunciation of a word while transforming it from a source script to a target script [1]. Transliteration from different languages to English is useful in bilingual knowledge extraction tasks including information retrieval, named entity recognition and automatic bilingual dictionary compilation. [2]-[4]. The out of vocabulary (OOV) words like names, and acronyms. In cross-lingual tasks are significantly transcribed into the base document language, provided that the source and target do not share the alphabet. The named entity transliteration plays a significant role in cross-language tasks, apparently, during document translation from source to target language, named entities are transliterated. Transliteration being the subtask of translation as it transforms one language script into corresponding similar phonetic characters of the target alphabet poses several challenges due to differences in syntax, morphology, and semantics between the source script and the target script language. Hindi to English transliteration or vice versa, pose dramatic challenges due to the morphologically rich nature of Hindi. For example, a Hindi word *चाभी* when transliterated into English has multiple transliterations of *chabhi*, *chaabhi*, *chaabhee*, *chaabhie*. Perhaps, the back transliteration is even more challenging as several words transliterate into a single target word. In this work, we employ the neural framework for transliteration, basically the sequence-to-sequence modelling based on recurrent neural networks (RNN) that are significantly popular in a wide range of tasks such as text summarization, machine translation, and named entity recognition [5], [6].

Transliteration has been studied for long, originally [7] modelled as a probabilistic finite state transducer machine, which was, subsequently, improved using phonetic and morphological models by [8]. Chinese tokens were generated by mapping English names to phonemes and then mapping each phoneme to the corresponding token [9]. An English-Russian transliteration system based on weighted finite-state transducer techniques and hidden Markov models was developed in [10]. A system for parallel Wikipedia titles in English to Tamil, English to Hindi, English to Arabic, and English to Russian using generative reinforcement models to produce mappings between source and target alphabet sequences is developed in [11]. A machine transliteration system for Bengali to English which relied upon the mapping of alphabets for each pair of Bengali-English phonemic mappings was designed in [12]. A phrase-based statistical machine translation model for English to Devanagari transliteration was proposed in [5]. They developed two distinct statistical systems using MOSES and Stanford Phrasal using English Hindi parallel corpus. Several researchers employed machine learning approaches such as [13] propounded transliteration of Marathi to English and Hindi to English named entity by segmentation of the source tokens into phonetic tokens and applying Support Vector Machines. The technique of Latin-to-Balinese script transliteration using a mobile application is interestingly critical [14]. Conditional random fields (CRFs) for transliteration of Hindi-English for cross-language information retrieval are suggested in [15]. CRF is also applied in a subword based approach to English to Indic languages (Hindi, Kannada and Tamil) [16]. CRF on the English to Korean transliteration and Hindi-English names respectively is suggested in [17]. A transliteration scheme that involved English to Hindi language pair from news 2009 transliteration task dataset is in [18]. The methodology incorporated English and Hindi contextual information for calculating the probabilities and chose the one which has a maximum probability and further improved the algorithm by applying post-processing rules. Josan and Kaur [19] suggested the transliteration techniques for Punjabi-Hindi in respect of Gurumukhi-Devanagari scripts by integrating the character level alignment from source vocabulary to target alphabets with statistical techniques. English to Chinese transliteration used a stack of convolutional network layers with a basic recurrent network layer on top, which produced promising output but still fell short of the phrase-based system of statistical machine translation [20]. Neural machine transliteration gained importance recently due to advancements in deep learning techniques [21]-[26].

The deep neural network (DNN) proved to be quite successful in several language processing tasks, however, less work is found in the literature on the problem of Hindi to English transliteration. Therefore, we investigate the effectiveness of deep learning models in transliteration by using an encoder-decoder based sequence to sequence model. As a preliminary task, we chose the gated recurrent units (GRU) networks as the basic element to design the encoder and decoder. The proposed neural machine transliteration framework which is essentially an encoder-decoder framework can produce more accurate transliteration than statistical systems by capturing the context of the source. The encoder converts the source word into a latent variable that holds the meaningful information which is subsequently, processed by the decoder to produce the transliteration word. The encoders and decoders are stacked with successive gated recurrent unit (GRU) layers on top of the input layer which handles the representation of the transliteration tokens which are individual characters. The character-level models are found more successful in sequence-to-sequence models. Therefore, for our work, we chose characters instead of words as the atomic elements used in the whole transliteration process. The contribution of this work is i) we experimentally evaluate the sequence to sequence neural architecture for English to Hindi and Hindi architecture and vice-versa using the parallel transliteration corpus and ii) we present the empirical results comparing one, two and three layers of GRU architecture for the same source and target scripts.

2. RESEARCH METHOD

2.1. Corpus

For the transliteration task we adopted the Hindi transliteration dataset of [27] in which 83,697 Hindi-English transliteration pairs are present. No multiword are present in the dataset. Corpus statistics are shown in Table 1.

Table 1. Corpus statistics

	Hindi	English
Maximum word size	25	28
Average word size	7.96	7.58
Total number of words	83697	83697

2.2. Proposed method

The sequence-to-sequence neural network modelling is a prominent technique which is based on the prediction of output sequence corresponding to its input sequence [28]. Transliteration can be viewed as an architecture analogous to the translation sequence to sequence neural model, moreover, it is a subtask in language translation while essentially useful in dealing with the named entities, which do not require translation [29]. This transliteration model is based on the concept of the encoder-decoder methodology which works well in many sequence-to-sequence applications [30], [31]. There are primarily two components namely, encoder and decoder, which are a sequence of connected layers. The encoder maps the input text to the fixed-size vector, which is the summarization of the source text, and this vector is given to the decoder to predict the sequence of generated characters. Both encoder and decoder are two-step phenomena to convert input words into a vector of floating-point numbers. In the first step, the text is converted into tokens of integers, whereas in the second step, such tokens into the matrixfloating-pointoint numbers with the help of an embedding layer. The overall transliteration process using the deep learning encoder-decoder method is illustrated in Figure 1. Yao *et al.* [32] has argued that the notion of encoder and decoder architecture is appropriate for general sequence to sequence models. The key principle is to map the entire input sequence to a vector and stack the layers of the GRU to produce a sequence of output based on the encoded vector.

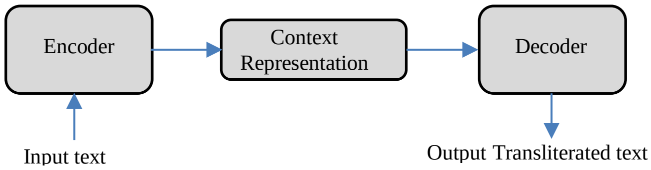

Figure 1. Transliteration using encoder decoder method

2.3. Gated recurrent units

The gated recurrent units (GRU) [33] are a type of recurrent neural networks (RNN) with the dedicated mechanism of resetting and updating the hidden state achieved using the reset gate and update gate respectively. The reset gate helps to control the amount of previous state that needs to be retained. Likewise, the update gate helps to control how much the new state gets from the old state. Both the gates are represented using (1) and (2). GRU is more streamlined and offers faster computations with a simplistic model among the RNN variants [33].

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{1}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{2}$$

where W and b denote weights and biases respectively. The output of the reset gate is integrated with the previous hidden state to obtain the intermediate current hidden state $\widetilde{H_t}$, which is further integrated with the update gate to obtain the final hidden state $H_t$as shown in:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \widetilde{H_t} \tag{3}$$

where the intermediate hidden state is given as:

$$\widetilde{H_t} = tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h \tag{4}$$

from (3) it can be concluded that with each GRU the short-term dependencies are controlled by the reset gates whereas the long-term dependencies are controlled by the update gates.

2.4. Proposed transliteration model

The information required by the transliteration system is contained in the words composed of characters. Moreover, each character contributes in a different manner. All the characters in Hindi contribute to the pronunciation, however, in English some are silent. Therefore, in the input layer and output layers, we tokenize the word to capture all the characters from the word. The words written in Hindi are morphologically rich and orthographically complex in nature. Each phoneme in Hindi may be weakly represented with a single character, however, English phonemes are sometimes composed of multiple alphabets. Therefore, when चाभी is transliterated to *chabhi*, च corresponds to ch, भ to *bh*. Hence, character level misalignments are abundant in Hindi to English. In our work, we consider each character as an individual unit for transliteration. Each input and output word pair is therefore tokenized into character level units instead of phonetic units. A dictionary is generated by assigning the highest integer value to the character with the highest frequency in the whole corpus. This dictionary is generated for both Hindi and English characters. Subsequently, each character unit is encoded with an integer value and, finally, the length of the encoded-word is normalized with padding. The post-processing technique adopted is the inverse of pre-processing. After encoding the vectors are obtained for each input word, and subsequently, Hindi words are passed into encoder and decoder, whereas English words are passed into decoder. Due to variation in length of words, the beginning and end markers are inserted at the input and output vector respectively, to align the vectors to the same length. Therefore, the size of the Hindi input vector obtained is 25 and the size for output vector obtained is 28.

The encoder with GRU layers accepts a varying dimension sequence as the source and converts it into the hidden layer of fixed dimension, following the design hypothesis of the encoder-decoder. In other words, the hidden state of the recurrent encoder captures the input sequence information. A decoder with the same number of GRU layers is utilized to predict the next token in order to produce the target sequence token by token based on which characters have been seen, alongside the source sequence recorded information. Figure 2 demonstrates how to use multiple GRU layers in transliteration for sequence-to-sequence training. The encoder's function is to convert a varying length source sequence into a context representation variable c of fixed shape and compress the source sequence information in it.

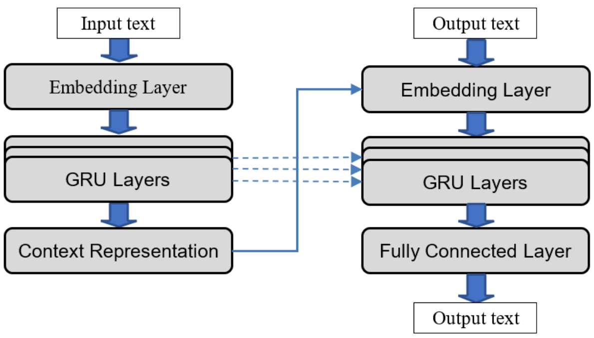

Figure 2. Proposed transliteration model

Let $x_1, \ldots, x_t$ be the source sequence where x_t being the t^{th} token in the source sequence. The recurrence converts the input vector x_t along with the preceding latent state h_{t-1} into the present hidden state H_t at the time step t. The encoder converts all the hidden states into the context representation at all time steps using specialised function q:

$$c = q(H_1, \ldots, H_T) \tag{5}$$

consequently, the context variable turn into just the final hidden state of the source sequence at the last step:

$$q(H_1, \ldots, H_T) = H_T \tag{6}$$

in our case, we focussed on bidirectional GRU, in which, the hidden state is based on the subsequence before and afterwards with respect to the time, including the present time step input, and transforms the entire sequence. Note that in order to acquire the compressed vector representation for each token collectively in the source sequence, we use the embedding layer. The embedding layer is significantly a set of weight matrices whose count of horizontal values is equal to the size of the source vocab and the count of vertical values is equal to the length of the context vector. For a given input token *i*, the embedding layer brings the i^{th} row of its weight matrix and returns it as the feature vector. The context variable c of the output of the encoder codes the complete source sequence x1,..., xT. Similarly, for a specified target sequence $y_1, \ldots, y_{T'}$ at every timestep t' the conditional probability for the decoder target subsequence and the context variable c is given as:

$$P(y_{t'} \mid y_1, \dots, y_{t'-1}, c) \tag{7}$$

it may be noted that the source and target sequences are of different lengths, so we use here t' for times step of output sequences in order to differentiate it with times step t of input sequences in encoders.

We use another layer of GRU as the decoder in order to model this probability conditioned on sequences. The GRU takes the output $y_{t'-1}$ from the previous time step at any time step t' of the output sequence and the context vector c as its input, and converts them along with the preceding hidden state $s_{t'-1}$ to obtain the current hidden state $s_{t'}$ during the present step. Consequently, the hidden layer transformation of the decoder as may be expressed as:

$$s_{t'} = g(y_{t'}, c, s_{t'-1}) \tag{8}$$

We explicitly utilize the final state in the last layer of the encoder to prepare the initialized decoder state. This enforces that there must be the same number of hidden units and layers in the encoder and the decoder GRU layers. The context representation variable is added in the decoder input at all the time steps to further enhance the encoded input sequence information. Subsequently, the fully connected layer is employed to convert the recurrent state to the last decoder layer to produce the likelihood of the target token. The decoder predicts a probability allocation for the target tokens at each time level. To get the distribution, we apply softmax and measure the cross-entropy loss for optimization. The separate padding tokens are added to the last of each sequence so that sequence of tokens having different lengths may be given as mini-batches in similar form. The predictions of the padded tokens, however, must be omitted.

3. RESULTS AND DISCUSSION

Using the appropriate neural network parameters the models are trained and tested with 10:1 split ratio and the validations accuracies are recorded. The best accuracy obtained among all the models is 92.3%. The metrics which are used to evaluate the predicted output are character error rate (CER) and word error rate (WER). The CER is defined as the fraction of correctly predicted characters among the total number of true characters, averaged over all the test words. The WER is defined as the fraction of correctly predicted words among the total test words. We present in Table 2 the CER and WER respectively, for the three models of Hindi to English transliteration models and three models of English to Hindi transliteration. The results show that on increasing the number of GRU layers there is improvement in both CER and WER. For Hindi to English transliteration, one and two additional layers improve the CER by 12.3% and 20.2% respectively, and WER by 1.3% and 6.2% respectively with respect to the single-layer model. On the other hand, the English to Hindi transliteration, one and two additional layers improve the CER by 13.3% and 26.1% respectively, and WER by 5.6% and 7.7% respectively when compared with the single layer model.

Table 2. Evaluation results of proposed transliteration models

Transliteration source and target language		*Hindi to English*		*English to Hindi*	
		CER	WER	CER	WER
Number of recurrent layers	*1*	0.253	0.691	0.398	0.848
	2	0.222	0.682	0.345	0.801
	3	0.201	0.648	0.294	0.783

We compare the proposed models with the recent existing works of Hindi to English transliteration in Table 3. Hindi to English transliteration includes works of [13], [15], and [17] have reported only the model network validation accuracy which is lower than the proposed work. We also observe better performance when compared with Arabic to English transliteration of [4], [29]. However, Chinese to English transliteration of [20] reports 3.1% lower, and Arabic to English transliteration of [21] reports 1.6% lower CER than the proposed model. Moreover, [24] reports the top-1 transliteration accuracy of 53.3%, which is equivalent to WER which is lower in this case.

We initially developed the models for Hindi to English transliteration, however, we also evaluated it over English as the source and Hindi as the target language. The comparison with existing works is presented in Table 4. This Hindi to English transliteration model shows better results with only some of the existing works only. In fact, CER is better than the English to Vietnamese model of [25].

Table 3. Comparison of proposed model with existing works on Hindi to English transliteration

Technique	Language pairs	Efficiency Rate
Cross connected multi-layer GRU	*Hindi to English*	ACC: 81.6 %; WER: 64.8%;CER: 20.1%
Orthographic similarity [23]	Tamil to English	ACC: 53.3%
GRU [4]	Arabic to English	WER: 81%
CNN + RNN [20]	Chinese to English	CER:16.2%
GRU + Attention [29]	Arabic to English	WER: 77.1%
CRF [17]	*Hindi to English*	ACC: 83.98%
SVM [13]	*Hindi to English*	86.52
DBN [21]	Arabic to English	CER: 22.7%
HMM [15]	*Hindi to English*	72.1%

Table 4. Comparison of the proposed model with existing works on English to Hindi transliteration

Authors	Language Pair	Efficiency Rate
Cross connected multi-layer GRU	*English to Hindi*	ACC: 70.6 %; WER: 78.3%; CER: 29.4%
Grapheme-phoneme [23]	English to Kannada	ACC: 85.93%
GRU + BiGRU [30]	English to Arabizi	ACC: 80.6%
LSTM + Attention [25]	English to Vietnamese	CER: 32.4%
GRU + Attention [30]	English to Arabic	WER:65.1%
CNN + RNN [20]	English to Chinese	ACC: 28.1%
Graph Reinforcement [11]	*English to Hindi*	F1: 93%
MEMM + Alignment [21]	English to Persian	ACC: 58.4%
HMM + WFST [11]	English to Russian	61%
CRF [16]	*English to Hindi*	41.8%

4. CONCLUSION

We specifically prepare models for the Hindi to English transliteration which is rarely addressed in the literature. The models are developed using the sequence to sequence neural network with an underlying encoder-decoder methodology. The GRU is used as recurrent units due to their simplicity and faster performance. The encoder translates the input text into an intermediate representation which is given to the decoder which maps the sequence with the output text. The character level approach is used for input and output representation for capturing subword level information. Different variants of the models are generated in single and multiple layers of GRU and the results are recorded. A maximum of 20.2 % improvement is observed in CER as compared to the base model in Hindi to English transliteration. It is observed that the performance improved on increases on increasing GRU layers is however at the cost of increased training time due to an increase in the number of parameters. We compare our work with existing Hindi to English, Arabic to English and Chinese to English transliteration models and observe that our model outperforms all with CER of 20.1% and validation accuracy of 81.6% excefor pt Chinese to English in which CER is 16.2%. We also apply the same model to devethe lop English to Hindi transliteration model by exchanging the input with output and vice-versa. However, their test results are not as good as the Hindi to English transliteration model.

REFERENCES

[1] A. Ekbal and S. Bandyopadhyay, “Transliteration of named entity: Bengali and English as case study,” in *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2007*, 2007, pp. 223–228.

[2] P. Virga and S. Khudanpur, “Transliteration of proper names in cross-lingual information retrieval,” in *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition -*, 2003, vol. 15, doi: 10.3115/1119384.1119392.

[3] K. Darwish, “Named entity recognition using cross-lingual resources: Arabic as an example,” in *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2013, vol. 1, pp. 1558–1567.

[4] Y. Merhav and S. Ash, “Design challenges in named entity transliteration,” *in COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings*, 2018, pp. 630–640.

[5] S. Sharma, N. Bora, and M. Halder, “English-Hindi transliteration using statistical machine translation in different notation,” in *International Conference on Computing and Control Engineering (ICCCE)*, 2012, no. Iccce, pp. 12–14.

[6] M. Z. Ansari, T. Ahmad, and M. A. Ali, “Cross script Hindi English NER corpus from Wikipedia,” in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 26, 2019, pp. 1006–1012, doi: 10.1007/978-3-030-03146-6_116.

[7] B. G. Stalls and K. Knight, “Translating names and technical terms in Arabic text,” in *Proceedings of the Workshop on Computational Approaches to Semitic Languages - Semitic ’98*, 1998, p. 34, doi: 10.3115/1621753.1621760.

[8] Y. Al-Onaizan and K. Knight, “Translating named entities using monolingual and bilingual resources,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL ’02*, 2001, p. 400, doi: 10.3115/1073083.1073150.

[9] S. Wan and C. M. Verspoor, “Automatic English-Chinese name transliteration for development of multilingual resources,” in *Proceedings of the 36th annual meeting on Association for Computational Linguistics -*, 1998, vol. 2, pp. 1352–1356, doi: 10.3115/980691.980789.

[10] P. Nabende, “Transliteration system using pair HMM with weighted FSTs,” in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration - NEWS ’09*, 2009, p. 100, doi: 10.3115/1699705.1699731.

[11] A. El-Kahky, K. Darwish, A. S. Aldein, M. A. El-Wahab, A. Hefny, and W. Ammar, "Improved transliteration mining using graph reinforcement," in *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2011, pp. 1384–1393.
[12] W. I. Net, D. Sen, and K. D. Garg, "A review on Bengali to English machine transliteration system," *International Journal of Software and Web Sciences*, pp. 60–64, 2015.
[13] P. H. Rathod, M. L. Dhore, and R. M. Dhore, "Hindi and Marathi to English machine transliteration using SVM," *International Journal on Natural Language Computing*, vol. 2, no. 4, pp. 55–71, Aug. 2013, doi: 10.5121/ijnlc.2013.2404.
[14] I. N. Jampel, G. Indrawan, and I. W. Widiana, "Accuracy analysis of latin-to-Balinese script transliteration method," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, p. 1788, 2018, doi: 10.11591/ijece.v8i3.pp1788-1797.
[15] S. Ganesh, S. Harsha, P. Pingali, and V. Varma, "Statistical transliteration for cross langauge information retrieval using HMM alignment and CRF," in *CLIA 2008 - 2nd International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies - 3rd International Joint Conference on Natural Language Processing, IJCNLP 2008 - Proceedings of the Workshop*, 2008, no. June, pp. 42–47.
[16] S. Reddy and S. Waxmonsky, "Substring-based transliteration with conditional random fields," in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration - NEWS '09*, 2009, p. 92, doi: 10.3115/1699705.1699729.
[17] M. LDhore, S. K Dixit, and T. D Sonwalkar, "Hindi to English machine transliteration of named entities using conditional random fields," *International Journal of Computer Applications*, vol. 48, no. 23, pp. 31–37, Jul. 2012, doi: 10.5120/7522-0624.
[18] A. Das, A. Ekbal, T. Mandal, and S. Bandyopadhyay, "English to Hindi machine transliteration system at NEWS 2009," in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration - NEWS '09*, 2009, p. 80, doi: 10.3115/1699705.1699726.
[19] G. Josan and J. Kaur, "Punjabi to Hindi statistical machine transliteration," *International Journal of Information Technology*, vol. 4, no. 2, pp. 459–463, 2011, [Online]. Available: http://www.csjournals.com/IJITKM/PDF 4-2/Article_26.pdf.
[20] Y. Shao and J. Nivre, "Applying neural networks to english-chinese named entity transliteration," in *Proceedings of NEWS 2016: 6th Named Entity Workshop at the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, 2016, pp. 73–77, doi: 10.18653/v1/w16-2710.
[21] T. Deselaers, S. Hasan, O. Bender, and H. Ney, "A deep learning approach to machine transliteration," in *Proceedings of the Fourth Workshop on Statistical Machine Translation - StatMT '09*, 2009, p. 233, doi: 10.3115/1626431.1626476.
[22] N. M. Nejad and S. Khadivi, "An unsupervised alignment model for sequence labeling: Application to name transliteration," in *2011 Named Entities Workshop*, 2011, p. 73.
[23] B. S. S. Lakshmi and B. R. Shambhavi, "An ensemble of grapheme and phoneme-based models for automatic English to Kannada back-transliteration," *International Journal of Intelligence and Sustainable Computing*, vol. 1, 2021, doi: 10.1504/ijisc.2021.113314.
[24] A. Kunchukuttan, S. Jain, and R. Kejriwal, "A Large-scale evaluation of neural machine transliteration for Indic languages," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 3469–3475, doi: 10.18653/v1/2021.eacl-main.303.
[25] N. T. Le, F. Sadat, L. Menard, and L. Menard, "Low-resource machine transliteration using recurrent neural networks," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 2, 2019, doi: 10.1145/3265752.
[26] B. Talafha, A. Abuammar, and M. Al-Ayyoub, "Atar: Attention-based LSTM for Arabizi transliteration," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, p. 2327, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2327-2334.
[27] M. M. Khapra, A. Ramanathan, A. Kunchukuttan, K. Visweswariah, and P. Bhattacharyya, "When transliteration met crowdsourcing: An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control," in *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, 2014, pp. 196–202.
[28] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to Sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, no. January, pp. 3104–3112, 2014.
[29] M. Rosca and T. Breuel, "Sequence-to-sequence neural network models for transliteration*," arXiv preprint arXiv:1610.09565*, Oct. 2016, [Online]. Available: http://arxiv.org/abs/1610.09565.
[30] M. S. Hadj Ameur, F. Meziane, and A. Guessoum, "Arabic machine transliteration using an attention-based encoder-decoder model," *Procedia Computer Science*, vol. 117, pp. 287–297, 2017, doi: 10.1016/j.procs.2017.10.120.
[31] A. Shazal, A. Usman, and N. Habash, "A unified model for {A}rabizi detection and transliteration using sequence-to-sequence models," in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, 2020, pp. 167–177, [Online]. Available: https://aclanthology.org/2020.wanlp-1.15.
[32] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding," in *Interspeech 2013*, Aug. 2013, pp. 2524–2528, doi: 10.21437/Interspeech.2013-569.
[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.3555.
[34] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111, doi: 10.3115/v1/w14-4012.

BIOGRAPHIES OF AUTHORS

Dr. Mohd Zeeshan Ansari is Assistant Professor at Department of Computer Engineering, Jamia Millia Islamia (A Central University), New Delhi. He completed his Ph.D. in Computer Engineering from Jamia Millia Islamia, New Delhi. He received M.Tech in Computer Science and Engineering from Delhi Technological University, New Delhi and B.Tech in Computer Science and Engineering from Uttar Pradesh Technical University, Lucknow. His area of research is Natural Language Processing, Code Mixing in Information Extraction, Text Mining and Deep Learning. His research currently focuses on developing multilingual natural language processing and information extraction models for social media text using with emphasis on supervised learning. He is active in participation of several national and international seminars, workshops and published articles in conferences and refereed journals. He is also reviewer of refereed journals. He can be contacted at mzansari@jmi.ac.in.

Dr. Tanvir Ahmad is currently Professor in the Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi. He received his Engineering degree from Bangalore University with First Class, M.Tech from I.P. University, New Delhi and Ph.D. from Jamia Millia Islamia, New Delhi. He has more than twenty five years of academic, research, training and administrative experiences in the field of Computer Engineering. He is the member of several academic and administrative bodies. His areas of research are Big Data Analytics, Text Mining, Graph Mining and Natural Language Processing. He can be contacted at tahmad2@jmi.ac.in.

Prof. Mirza Mohd Sufyan Beg obtained his B.Tech. (Electronics) degree from the Aligarh Muslim University, India in 1992 with First Rank. He obtained his M.Tech. (Microelectronics) degree from IIT Kanpur, India in 1994. Thereafter, he joined the Department of Electronics Engineering, Aligarh Muslim University, India as a member of the faculty. He has been a Lecturer, a Senior Lecturer and then a Reader in the Department of Computer Engineering at the same University. While on study leave from there, he obtained his Ph.D. degree in the area of Computer Technology from IIT Delhi, India in 2004. He also visited the University of California at Berkeley as a BT Fellow. Currently, he is a Professor and Chairman at the Department of Computer Engineering at the Aligarh Muslim University. He is also serving as the Principal of Z. H. College of Engineering and Technology since August 2015. He can be contacted at email: mmsbeg@hotmail.com.

Dr. Faiyaz Ahmad Assistant Professor Department of Computer Engineering, Jamia Millia Islamia, New Delhi has completed B.Tech in Computer Engineering from Institute of Engineering & Technology, Agra and M.Tech in Information Technology from Guru Gobind Singh Indraprastha University Delhi. He optained Ph.D. in Computer Engineering from Jamia Millia Islamia, New Delhi. His area of research interest is image information retrieval, image mining, medical image analysis and recogntion. He has teaching experience of more than 14 years. He has published number of high quality research papers and articles in various reputed Journals and National and International Conferences. He can be contacted at fahmad1@jmi.ac.in.