Deep Learning based Roman-Urdu to Urdu Transliteration

Article *in* International Journal of Pattern Recognition and Artificial Intelligence · September 2020

DOI: 10.1142/S0218001421520017

2 authors, including:

Mehren Alam
National University of Computer and Emerging Sciences
9 PUBLICATIONS **79** CITATIONS

SEE PROFILE

Deep Learning based Roman-Urdu to Urdu Transliteration

Mehreen Alam

Computer Science Department
National University of Computer and Emerging Sciences
Islamabad, Pakistan
mehreen.alam@nu.edu.pk

Sibt ul Hussain

Computer Science Department
National University of Computer and Emerging Sciences
Islamabad, Pakistan
sibtul.hussain@nu.edu.pk

Attention based encoder-decoder models have superseded conventional techniques due to their unmatched performance on many neural machine translation problems. Usually, the encoders and decoders are two recurrent neural networks where the decoder is directed to focus on relevant parts of the source language using attention mechanism. This data-driven approach leads to generic and scalable solutions with no reliance on manual hand-crafted features. To the best of our knowledge, none of the modern machine translation approaches have been applied to address the research problem of Urdu machine transliteration. Ours is the first attempt to apply the deep neural network based encoder-decoder using attention mechanism to address the aforementioned problem using Roman-Urdu and Urdu parallel corpus. To this end, we present i) the first ever Roman-Urdu to Urdu parallel corpus of 1.1 million sentences, ii) three state of the art encoder-decoder models, and iii) a detailed empirical analysis of these three models on the Roman-Urdu to Urdu parallel corpus. Overall, attention based model gives state-of-the-art performance with the benchmark of 70 BLEU score. Our qualitative experimental evaluation shows that our models generate coherent transliterations which are grammatically and logically correct.

Keywords: deep learning; Roman-Urdu to Urdu transliteration; attention models; neural machine translation; natural language processing.

1. Introduction

Transliteration is the process of converting text from one script to another with no change in meaning, that is, mapping only the letters in the source language to the letters of the target language. Since the focus is on pronunciation only, it makes a language more accessible to people who are unfamiliar with the script. It can also be treated as a translation problem as they share major challenges. The challenges include being able to address the differences between the source and target languages in context of syntax, semantics, morphology, and alignment.

Techniques used to address this problem have previously relied on phrase-based, rule-based, and statistics-based methods.[28,60,65] However, these techniques have

been superseded by the neural network based machine translation methods (NMT) as the translation quality produced by the latter nearly approaches the quality of a human translation.[65] NMT does not rely on hand-crafted features and explicit annotation and is, therefore, able to produce results that are more scalable and generic, thus being well-equipped to handle the complexities inherent in the languages. Specifically for machine transliteration, the use of neural network based methods is still in its inception with the exception of few works.[22,30,50] To the best of our knowledge, no significant effort has been done to apply any neural machine translation (NMT) method to address the issue of Roman-Urdu to Urdu transliteration. In this paper, we first formulate neural network based Roman-Urdu to Urdu transliteration methods and then present a detailed empirical analysis on these methods.

One of the most popular methods to solve this problem is the encoder-decoder based sequence to sequence model, where encoders and decoders are two separate networks built using two Recurrent Neural Networks (RNNs).[11] The encoder maps a variable-length source sequence onto a fixed-length latent vector while the decoder maps the vector representation back to a variable-length target sequence. The two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence.[11] For machine translation, the source sentences are fed to the encoder network and target sentences are fed to the decoder network during training time. The length of both networks or languages need not be same, which facilitates the capture of inherent heterogeneity of both the languages. The two languages may differ from one another in various aspects. For example, language direction being right-to-left or left-to-right, degree of morphological richness, vocabulary size, or alignment. For our work, we have chosen long short-term memory (LSTM) based cells for our encoder and decoder network as they are known to overcome the problem of vanishing gradients,[20,57] and can also capture longer dependencies in the sentence.

Inspired from the works of Refs. 58, 8, we use our custom modified versions of encoder-decoder based models after rigorous experimentation and parameter tuning. We eventually produce three models of varying complexity and performance. These models are:

(1) basic encoder-decoder model,
(2) bidirectional encoder-decoder model, and
(3) bidirectional encoder-decoder model with attention mechanism.

The first model is the vanilla encoder-decoder network illustrated in Figure 1. The second model uses bidirectional encoder which learns from the words occurring in both the time dimensions, i.e. from the past as well as the future (Figure 2). The third model is an extension of the second model with the addition of the attention mechanism,[7,8,63] which lets the decoder focus on the relevant part of the input words when predicting the target word (Figure 3). Both encoder and decoder

networks are deep LSTM networks to better capture the structural, sequential and contextual dependencies with their architecture details discussed in Section 3.

1.1. Challenges

The main challenge of any transliteration task is to produce output sentences that are grammatically correct and make sense to humans. In addition to these basic requirements, the following complexities need to be addressed:

(1) To give accurate transliteration for short to medium length sentences, models need to handle the context. The complexity of remembering the context gets exacerbated with the increase in sentence length.
(2) Models need to keep contextual information in view while transliterating homophones. For example, the Roman-Urdu word "aam" has two distinct meanings depending on the context it is used in: عام or آم. Another example could be to pick the correct verb from its conjugations using the contextual information, i.e. transliteration for the sentence "hum school gae" shall be ہم اسکول گئے or ہم اسکول گئی.
(3) Models need the ability to handle varying length sentences, i.e., giving the transliteration for even those sentences where the source and target sentence lengths may not be the same. For example, for an Urdu sentence like یہ اسلام آباد ہے with 4 individual words, the transliterated Roman-Urdu sentence has three words "Yeah Islamabad hai." Similarly, for the Roman-Urdu sentence "Kis ke lye hai?" with 4 individual words, the Urdu script has three corresponding words کس کیلئے ہے؟

1.2. Contributions

Overall, we make the following contributions:

(1) We design the first parallel corpus of Roman-Urdu to Urdu mapping of 1.1 million sentences that we also release publicly.[a]
(2) We propose three different variants of state-of-the-art encoder-decoder models: model with unidirectional encoder, model with bidirectional encoder, and model with bidirectional encoder using attention mechanism. The latter sets the state of the art with a BLEU score of 70 on Roman-Urdu to Urdu machine transliteration task.
(3) Our detailed empirical analysis of the three above mentioned encoder-decoder networks for machine transliteration task demonstrates that our models learn all relevant syntactic, semantic, and contextual information, and that they give us the translation quality which is very close to human translation.

[a]https://bit.ly/2MB5QXR

Rest of the paper is organized as follows. Section 2 discusses the related work in detail. We give an outline of all the three models' architectures in Section 3 followed by all the relevant details about the dataset in Section 4. Section 5 gives the experimental settings followed by results and discussions. Section 6 contains conclusions while future directions are discussed in Section 7. English translation of every word and sentence in Urdu or Roman-Urdu languages is given in Section Appendix A.

2. Related Work

Machine Translation is a problem in the domain of natural language processing where researchers have worked on finding automated ways of translating between natural human languages. This field emerged in 1949 in Warren Weaver's Memorandum of Translation. It is a very challenging task as no machine to date has achieved human parity to translate from one language to another.[18] The main challenge in this task arises when correlations have to be built on two languages that generally lack any standardization in many ways including linguistic morphology or topology, requiring much more than just mere substitution of words or phrases. Previously, the commonly used approaches for machine translation have been either rule-based, statistical-based, example-based, or a hybrid of these techniques.[65] The major drawback of all the approaches has been reliance on manual feature engineering and explicit annotations which led to lack of scalability and generality. Neural networks based machine translation systems known as the neural machine translation (NMT) take a generic and scalable approach, delivering performance that surpasses the performance of previously used machine translation methods. NMT techniques are also independent of any explicit annotation be it manual, automated, or semi-automated.

At the time of writing, deep learning is the most widely used technique to solve various research problems like computer vision, natural language processing, automatic speech recognition, and bio-informatics.[32,38] Specifically, for natural language processing tasks like machine translation,[65] language modeling,[54] sentiment analysis,[3,51,61] conversational modeling,[62] handwriting generation,[43] and text to speech conversion,[46,59] deep learning has outperformed all the conventional approaches to solving these problems. State-of-the-art deep learning models for machine translation include the encoder-decoder structure proposed by Ref. 25 in 2013, which was further enhanced by the sequence to sequence model proposed by Ref. 58 in 2014. The latter had an edge that the model relied entirely on Long Short-Term Memory Network[20] for both the encoder and decoder network and overcame the problem of long distance re-ordering and of vanishing/exploding gradients. This approach has the limitation that it relies on just one context vector that stores all the information from the encoder. Enhancements to these models were done by Refs. 8, 11 in 2014 where attention mechanisms were used to guide the decoder network to focus more on the relevant part of the input for better translation. Now for every target

word, the decoder predicts on the basis of dynamically generated context vectors from the input sequence. This has shifted the whole focus to the attention models as these models have led to further improvements in neural machine translation.[24]

Word Embeddings have also been used successfully as input to the encoders which is equivalent to feeding the network with richer representations instead of the context-less one-hot encodings. Many enhancements have been done on the vanilla Word2Vec model, which was developed by Ref. 37; it is an unsupervised predictive way of absorbing all the relations between words. Global vectors of word representation (Glove)[44] learns embeddings by capturing the count of overall statistics of how often a given word appears. FastText, proposed by Ref. 23, treats each word as a composition of n-gram characters which, though slower than its counterparts, is better at generating embeddings for rare words and out of vocabulary words.[13] Work on learning distributed representations has been extended to paragraphs,[31] sentences,[41] and topics.[40] Ref. 2 has applied the word2vec embeddings for experimentation to learn Roman-Urdu to Urdu transliteration.

Various techniques have been applied to improve performance of the neural networks. Residual connections have been applied to LSTMs in encoder-decoder models since simply stacking layers is effective up to a certain depth.[10,55,64,65] Dropout, a good regularization method to prevent over-fitting,[9,56] has also been successfully used in the research areas of question answering,[66] image classification,[33] language modeling,[21] speech recognition,[45] unsupervised pre-training for sequence to sequence,[48] convolutional sequence to sequence,[15] and machine translation.[34] Apart from feeding the encoder with the source text in a straight forward way, improvements have also been observed by reversing the order and using the bidirectional encoder having both forward and backward orders at the same time.[8] Bidirectional encoders are able to capture dependencies of the past as well as of future inputs, while unidirectional encoders can only take past inputs into account.[10] Bidirectional decoders, proposed by Ref. 52, work by minimizing the error propagation of reverse target-side context modeling.

Urdu is a low-resource yet morphologically rich language, which means in Urdu it is possible that for a single word there may exist many variants. Use of deep learning to solve research problems in Urdu language is still in inception as it is evident in the areas of automatic speech recognition,[4] sentence boundary disambiguation,[47] sentiment analysis,[53] transfer learning,[67] and machine translation.[49] Specifically, machine translation problems in the context of Urdu language still use conventional phrase based or rule based methods.[5,12,19,39] In the domain of Roman-Urdu to Urdu transliteration, only the work done by Ref. 2 shows promising results using basic encoder-decoder networks despite using a small-scale dataset. None of the works in machine translation for Urdu language employs mechanisms like attention, bidirectional encoder, beam search, residual connection, drop-out, or bucketing.

Since deep learning techniques are data-driven, a large scale dataset is a necessary resource to effectively model the diversity and capture the inherent complexities of both the languages. Having a parallel corpus opens the avenues for further

research in the area. For instance, Ref. 27 provides parallel corpora in 11 languages including English, Dutch, Spanish, Italian, and Swedish, to name a few. Unfortunately, there is no publicly available large-scale parallel Roman-Urdu to Urdu corpus, which has been a major bottleneck in exploring further research opportunities in the domain of machine translation. Transliteration is generally treated as a sub-form of translation, as transliteration maps the letters from one alphabet into the corresponding similar-sounding characters of the target alphabet. Specifically, sequence to sequence models have been applied to the task of machine transliteration too. Ref. 1 transliterates from Sanskrit to English, while Ref. 36 transliterates from English to Persian using an attention based approach.

To the best of our knowledge, apart from the preliminary work done by Ref. 2, neural network based machine translation techniques have not been applied for Roman-Urdu to Urdu transliteration and no Roman-Urdu to Urdu parallel corpus is publicly available. Thus, this work tries to address these shortcomings by providing machine transliteration models and a large scale parallel corpus for Roman-Urdu to Urdu transliteration.

3. Model Architecture

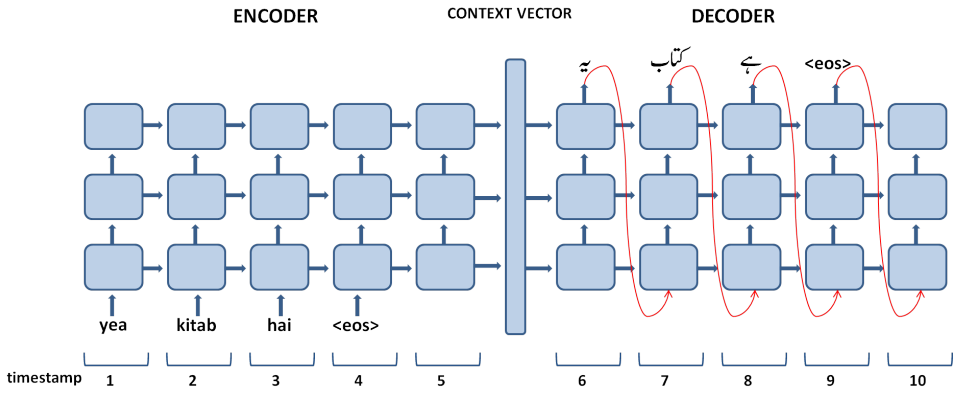

Fig. 1: Basic Encoder Decoder Model. Encoder and decoder networks are both three layer deep LSTM based RNNs while the context vector absorbs the embeddings from the encoder which are used by the decoder while predicting the word.

3.1. Basic Encoder Decoder Model

Our basic Encoder-Decoder model, as shown in Figure 1, is composed of two RNN based models, an encoder which embeds the learning of the source sentence of n words $x_1, x_2, ..., x_n$, into a context vector c, and a decoder that uses the learnt context vector to map input words to the target sentence of m words $y_1, y_2, ..., y_m$,

using conditional probabilities. Conditional probability of a target sentence y given input sentence x is written as

$$p(y|x) = \Pi_{j=1}^{m} p(y_j|y_{<j}, c), \tag{1}$$

while in practice, we use log probabilities, i.e.

$$\log p(y|x) = \sum_{j=1}^{m} \log p(y_j|y_{<j}, c), \tag{2}$$

such that

$$p(y_j|y_{<j}, c) = softmax(g(h_j)), \tag{3}$$

where probability of every word to be predicted y_j, given the previous words predicted $y_{<j}$, and the context vector c, is equal to taking softmax over g. Here g is a transformation function that maps the output vector to the size of output vocabulary and h_j is the LSTM output in the decoder network which is computed as

$$h_j = f_a(h_{j-1}, c), \tag{4}$$

where f_a is usually a nonlinearity such as tanh or ReLU. c is the hidden state learnt from the last time instance of the encoder LSTM, i.e. h_{n-1} and computed as

$$c = f_b(x_i, h_{n-1}), \tag{5}$$

where f_b is usually a nonlinearity such as tanh or ReLU and may not be the same as in Eq. (4). Refs. 58, 34 have used LSTM instead of GRU and vanilla RNN as it has shown to work better on longer sentences and on predicting rare words.

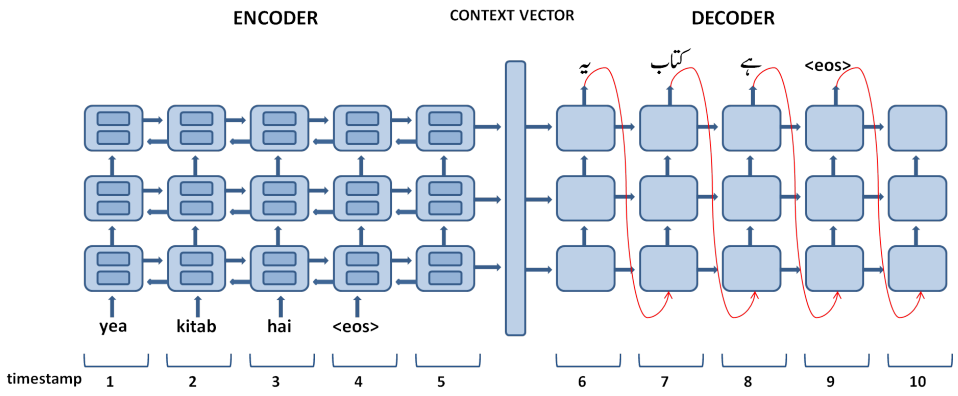

Fig. 2: Bidirectional Encoder Model. It is the the same as the vanilla encoder-decoder network with the addition that the encoder network is bidirectional, which means it learns in the forward as well as the backward direction.

3.2. Bidirectional Encoder and Basic Decoder Architecture

The encoder in the basic encoder-decoder network only learns from the context in the past, which can limit the extent to which it learns the context. The learning capacity can be enhanced by using a bidirectional encoder,[52] where the encoder learns in the forward as well as backward time direction using pairs of LSTMs, as shown in Figure 2. Now the output of every LSTM in the encoder network is as follows:

$$h_i = concat[\overrightarrow{h_{i+1}}; \overleftarrow{h_{i-1}}], \tag{6}$$

such that

$$\overrightarrow{h_{i+1}} = f_c(x_i, h_{i+1}), \tag{7}$$

$$\overleftarrow{h_{i-1}} = f_c(x_i, h_{i-1}), \tag{8}$$

where f_c usually is a nonlinearity such as tanh or ReLU and may not be the same as in Eq. (4) and Eq. (5).

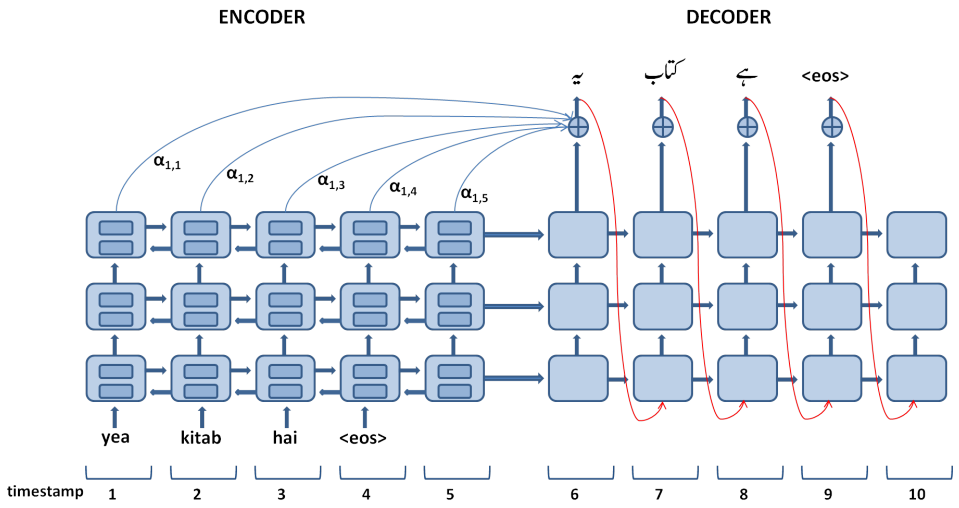

Fig. 3: Bidirectional Encoder with Attention Model. For clarity, only the process of making the context vector for the first output word is shown from the input words with corresponding weights.

3.3. Bidirectional Encoder, Attention Decoder and Attention Mechanism

Both the previous models only use one context vector for storing all the information and rely solely on it for decoding purposes. Our third model circumvents this by including an attention mechanism as shown in Figure 3, and is a combination of

a bidirectional encoder and an attention decoder based on Ref. 8. The attention model overcomes the bottleneck of having a single context vector holding all the information of the source sentence. Specifically, its decoder uses a separate context vector and a separate annotation vector for every input word while predicting every output word. This helps the decoder to focus more on the context relevant to the current word to be predicted rather than on the whole input sentence. The annotation vector h_i for every word x_i, is the combination of its forward annotation vectors $(\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_{i-1}})$ and the backward annotation vectors $(\overleftarrow{h_{i+1}}, \overleftarrow{h_{i+2}}, ..., \overleftarrow{h_n})$, c.f. Eq. (6). Each of these annotation vectors is then used to calculate the context vector c and alignment score α which quantifies how much the word at the i_{th} position in input matches with the word at the j_{th} position in the output sequence. Context vector c_j, for every output word y_j is defined as:

$$c_j = \sum_{j=1}^{n} \alpha_{ji} h_i, \tag{9}$$

where alignment score α_{ji} for each annotation vector h_i is calcualted as:

$$\alpha_{ji} = \exp(e_{ji}) / \sum_{k=1}^{n} \exp(e_{jk}). \tag{10}$$

In Bahdanau's model,[8] e_{ji} is calculated as:

$$e_{ji} = a(s_{j-1}, h_i), \tag{11}$$

where s_{j-1} is the output of the previous LSTM hidden unit just before emitting y_j, and a is alignment model, i.e.

$$a(s_{j-1}, h_i) = {V_a}^T tanh(W_a[s_{j-1}; h_i]), \tag{12}$$

where V_a^T and W_a are alignment model matrices to be learnt during the training.

4. Dataset

One of our major contributions is to develop 1.1 million sentence-pairs of parallel Roman-Urdu to Urdu corpus that we release publicly.[a] It was a challenging effort as the famous crowd-sourcing facility, Amazon Mechanical Turk (AMT), was not available in Pakistan. Even if it were available, the requirement for having such a large number of workers to be proficient in the Urdu language might have made it infeasible. Therefore, we had to rely on local sources to collect and curate a large dataset which contains the inherent diversity in language morphology, semantics, and the domains it comes from. We divide the data curation process into three major parts: crawling and scraping the data, generating a Roman-Urdu to Urdu parallel corpus, and checking quality of the parallel corpus, as explained below in sub-sections 4.1, 4.2, and 4.3 respectively.

4.1. Crawling and Scraping the Data

We crawled the web and downloaded Roman-Urdu and Urdu sentences from over 20 websites having articles, novels, history, blogs, posts, and news from categories of sports, politics, entertainment, business, and technology. We were able to raise a monolingual Urdu corpus of 1 million sentences and a monolingual Roman-Urdu corpus of 0.2 million sentences.

4.2. Generating a Roman-Urdu to Urdu Parallel Corpus

Due to the scale of the datasets, it was infeasible for us to manually cross-annotate both the collected datasets, which made a total of 1.2 million sentences (1 million Urdu and 0.2 million Roman-Urdu), i.e. getting manual translations from Roman-Urdu to Urdu and manual translations from Urdu to Roman-Urdu. Refs. 14, 29, 17 have approached this problem by getting computer-aided machine translation for the datasets and later involving human translators to post-edit the translated dataset. Such an approach is not only cost-effective but also scalable and leads to results in a reasonable time. Following them, we used the portal iJunoon[b] to transliterate Urdu sentences to Roman-Urdu and vice versa. This process led to a parallel corpus of Roman-Urdu to Urdu having 1.2 million sentences.

4.3. Quality Checks for Roman-Urdu to Urdu Parallel Corpus

Our huge parallel corpus still had many anomalies that were fixed using manual and automated techniques. For example,

(1) Sentences having a mix of Roman-Urdu and Urdu words were removed from both sides of the dataset, which could be because the word to be transliterated was either a rare word, a proper noun, or a word from a foreign language.
(2) Some of the sentences were unreasonably long or short to be useful for mapping onto a sequence to sequence task, and hence discarded.

Manual post-editing was done to check the sentences are logically and grammatically correct.

4.4. Dataset Details

We were finally able to generate a clean and aligned Roman-Urdu to Urdu parallel corpus of exactly 1,107,156 sentences. Our corpus has a total Urdu vocabulary of 34,523 words and a total Roman-Urdu vocabulary of 21,021 words, as shown in Table 1. It is interesting to note that Urdu vocabulary has more words than Roman-Urdu words. It is because in Urdu, there are many words which have a

[b]www.ijunoon.com

space between them but are considered one word only. Any such compound word in Urdu is usually represented as a single word in Roman-Urdu. For example, compound Urdu words like اسلام آباد, بے وقوف, and علم و ادب have their corresponding Roman-Urdu transliteration as a single word, that is, "Islamabad", "bewakuf", and "ilmoadab", respectively.

Table 1: Details of the parallel corpus, Roman-Urdu to Urdu.

Roman-Urdu to Urdu Corpus	1,107,156 Sentences
Total Roman-Urdu Words	21,021
Total Urdu Words	34,523

The dataset was randomly divided into train, development, and test sets by the ratios of 70%, 15%, and 15%, respectively. For input to our sequence to sequence models, we converted our parallel corpus into its indexed form. Every unique word was mapped to a number. For input, we gave an indexed form of Roman-Urdu words as input and got as output the indexed form of Urdu words that was converted back to the original Urdu script as explained in Table 2.

Table 2: Steps to show the transformations an input Roman-Urdu sentence goes through before getting the final answer in its human-readable Urdu form.

Step 1	Our Input	Sara aur Zara dost hain
Step 2	Indexed Roman-Urdu	21 52 1 664 3200
Step 3	Indexed Urdu	451 562 2343 44 5
Step 4	Converted Output	سارہ اور زارا دوست ہیں

It is worth noting that we are also considering unequal length sentences in Roman-Urdu and Urdu parallel corpus, so that we overcome the limitation of having one to one correspondence on the basis of alignment. However, we do not take into account the Urdu or Roman-Urdu short hand used for text messaging or in tweets. We have also not used any word embedding algorithm to map every word to its vector representation since the sequence to sequence model has the inherent capacity of learning the dependencies itself. We have already explored word embeddings in one of our previous works.[2]

5. Experimental Settings

We performed thorough experimentation on all the three above mentioned models using the parallel corpus of Roman-Urdu to Urdu. The corpus consists of 1.1 million

sentences, generated using a mix of automated and manual techniques. Our Roman-Urdu vocabulary is of 21K words while Urdu vocabulary is of 35K words. All of our experiments were run using Google's Tensorflow based tf-seq2seq library.[58] We used three layered deep LSTMs for both encoder and decoder networks with input and output sequence lengths capped at 40. Attention and context vector dimensions were fixed at 1024 after thorough experimentation in the range of 250 to 4096. This ensures that the model is large enough to absorb the correlation and it is of a reasonable size to be able to deliver output in a reasonable time. For optimization, Adam[26] with epsilon 10^{-7} is used. We initially used the learning rate of 0.0001, which was gradually reduced by a factor of 10 as the learning progressed. We used bucketing during training by dividing our train data into 5 buckets with sentence length of 0-10, 10-20, 20-30, 30-40 and > 40. We used Bilingual Evaluation Understudy (BLEU)[c] score proposed by Ref. 42 for evaluation. This is the standard evaluation metric to measure the quality of translation.[6,35,64,65] We conducted each experiment on a batch size of 64 and ran for 2 epochs (one epoch = 17,300 train steps, one train step = 64 examples). Cross validation was done after every 2000 train steps. For inference, we have used models with and without beam search[16] for neural machine translation. Finally, we used the model giving the highest BLEU score on the validation set for reporting results on the test set.

6. Results and Discussion

6.1. Quantitative Analysis

Our results show that the bidirectional model using attention gives the highest BLEU score improving upon the other two models' score by at least 10. It is also the quickest to converge as compared to the two other models showing very clearly that it has the capacity to learn accurately and efficiently. Figures 4 and 5 show the status of the BLEU score and loss respectively on validation data across all the train steps for all the three models. It can be seen that the bidirectional model using attention achieves the BLEU score of 70 after 30K train steps while the bidirectional encoder based model converges to the BLEU score of 60 after 36K train steps. Unidirectional encoder model can be seen to become stagnant, and it does not learn any more after 30K train steps.[d] This pattern of stagnation for BLEU score is also common for the other two models. Loss for the three models complements our inferences, where the bidirectional model with attention mechanisms hits its lowest on the train steps of 28K, while the other two models reach a plateau near 36K train steps. As it happened with the BLEU score, the bidirectional model using attention is the quickest to converge to the least loss value too. It also reacts to over-fitting more profoundly as compared to the other two models when trained

[c] `github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[d] using paired bootstrap resampling method, the models are 95% statistically significant than the other models.

beyond the optimum point as it is evident from the graph. BLEU score of 70 for the bidirectional model using attention is a remarkable achievement and it clearly implies that the model is generic, accurate, and reliable enough to be used for unseen data.

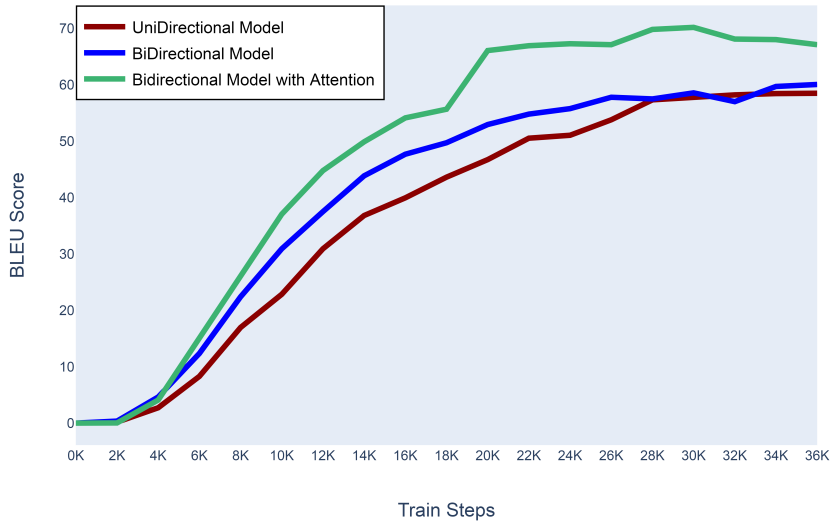

Fig. 4: Variation of BLEU score on the validation set for all the three models over training steps. The bidirectional model using attention is seen to achieve the highest BLEU score in the shortest time, i.e. BLEU score of 70 after 30K train steps. The bidirectional encoder based model reaches the BLEU score of 60 after 36K train steps while the uni-directional encoder model reaches the BLEU score of 58 and does not learn any more after 30K train steps.

Although all our three models give excellent performance, it is the bidirectional model using attention mechanism that beats both the other models by a total of 10 BLEU points. Second place is occupied by the bidirectional model without attention which outperforms the basic model by 2 BLEU points.

6.2. Qualitative Analysis

Although our models have given excellent BLEU score, qualitatively their performance is even more promising. The results are shared in Tables 3 and 4, and it is very encouraging to see that all the three models give a meaningful, logically coher-

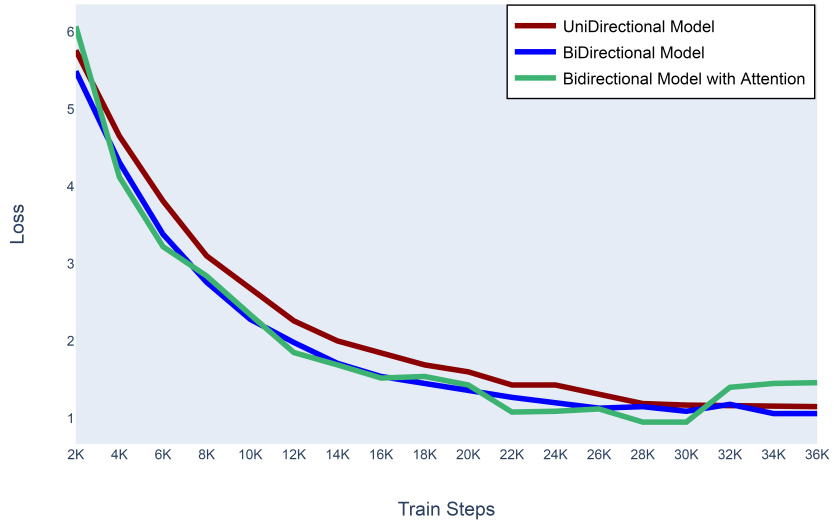

Fig. 5: Variation of loss on validation set for all the three models over training steps. Loss for the three models complements our inferences, where the bidirectional model with attention mechanism hits its lowest on the train steps of 28K, while the other two models reach a plateau near 36K train steps.

ent, and grammatically correct transliteration for short and long length sentences. The models have successfully captured the semantic and syntactic relationships between the source and the target language, which is why majority of the words transliterated are accurate and their flow is also correct. It is worth noting that all the three models perform much better on shorter sentences than on longer ones, which is expected as the context gets lost with the increase in the length of the sentence. Though the model with attention mechanism tends to perform better on both types of lengths.

We can also see interesting patterns when we compare the performance of all the three models. In Tables 3 and 4, we show two sentences which are entirely correctly transliterated and two which contain incorrect transliteration picked randomly from the test set, so that we get a precise representation of the whole test result. Incorrectly transliterated words are highlighted by underlining them. The observations presented in subsections 6.2.1 and 6.2.2 can be made from the results.

Table 3: Transliteration of Short Sentences. (Note that mistakes are highlighted via underlines).

		Sentences	Sentence No
Input:	Roman-Urdu Input	bohat shukria aap ka is izzat afzai ke liye	1
		bohat khobsorat tasaveer hai	2
		gussa mein kuch bhi ho sakta hai	3
		aaj ke din to bohat hi masroof rahi hon	4
		kuch ahbaab ka khayaal hai mein dr hon	5
		aashorh ke juloos par hamla hua	6
		Shukrai nazar Ahmed sahib ! ye nazam mein yahan bhi post kar chukka hon	7
Output:	Basic Encoder Decoder	بہت شکریہ آپ کا اس عزت افزائی کے لیے	1
		بہت خوبصورت تصاویرہیں	2
		غصہ میں کچھ بھی ہو سکتا ہے	3
		آج کے دن تو بہت ہی مصروف رہی ہوں	4
		کچھ احباب کا خیال ہے میں ڈاکٹر ہوں	5
		چونے کے جلوس پر حملہ ہوا	6
		شکریہ احمد صاحب ! یہ نظم میں بھی یہاں کو شاملہیں	7
	Bidirectional Encoder Decoder	بہت شکریہ آپ کا اس عزت افزائی کے لیے	1
		بہت خوبصورت تصاویرہیں	2
		غصہ میں کچھ بھی ہو سکتا ہے	3
		آج کے دن تو بہت ہی مصروف رہی ہوں	4
		کچھ احباب کا خیال ہے میں ڈاکٹر ہوں	5
		بجریکے جلوس پر حملہ ہوا	6
		شکریہ احمد صاحب ! یہ نظم میں بھی یہاں کو اٹھایاہوں	7
	Bidirectional Encoder Decoder with Attention	بہت شکریہ آپ کا اس عزت افزائی کے لیے	1
		بہت خوبصورت تصاویرہیں	2
		غصہ میں کچھ بھی ہو سکتا ہے	3
		آج کے دن تو بہت ہی مصروف رہی ہوں	4
		کچھ احباب کا خیال ہے میں ڈاکٹر ہوں	5
		چڑیوںکے جلوس پر حملہ ہوا	6
		شکریہ احمد صاحب ! یہ نظم میں بھی یہاں پوسٹ کرچکا ہوں	7
	Original Output	بہت شکریہ آپ کا اس عزت افزائی کے لیے	1
		بہت خوبصورت تصاویرہیں	2
		غصہ میں کچھ بھی ہو سکتا ہے	3
		آج کے دن تو بہت ہی مصروف رہی ہوں	4
		کچھ احباب کا خیال ہے میں ڈاکٹر ہوں	5
		عاشورہ کے جلوس پر حملہ ہوا	6
		شکریہ احمد صاحب ! یہ نظم میں بھی یہاں پوسٹ کرچکا ہوں	7

6.2.1. Short Length Sentences

It can be observed that all models hardly make any mistake on short length sentences, seen from Table 3. While the basic encoder-decoder and bidirectional encoder models do make a few mistakes, the performance of bidirectional encoder with attention model is exceptionally good. We can also observe the correctness in sentence construction, in the use of parts of speech, and in mapping of grammatical rules. Mistakes made by any of the three models are handled in intelligent ways, i.e. even the incorrect transliteration of words is done in ways such that the sentence still makes sense. For example, for the reference word عاشورہ in sentence 6 which is a noun, all the models replace it with a noun that makes the most sense چڑیوں, بجری, and چونے. Moreover, the attention based model correctly transliterates the rather uncommon verb phrase پوسٹ کرچکا in sentence 7, while the other two models do make a mistake but in such a way that the flow of the sentence is not broken and the phrases with the maximum probabilities chosen are کو اٹھایا and کو شامل.

Table 4: Transliteration of Long Sentences. (Note that mistakes are highlighted via underlines).

			Sentence No
Input	Roman-Urdu Input	kal mein ne barri mushkil se english mein tabdeel kya lekin aap ko tag phir bhi nahi hwa	1
		is terhan aap jald shtyab ho satke hain aur ghar mein doosron ke bemaar honay ka khadsha bhi kam ho ga	2
		aindah jaane ka programme abhi banaya nahi hai baqi Allah behtar jaanta hai	3
Output	Basic Encoder Decoder	کل میں نے بڑی مشکل سے انگلش میں تبدیل کیا لیکن آپ کو ٹیگ پھر بھی نہیں ہوا	1
		اس طرح آپ جلد کریں دے مقصد اتنا سے کرنے کرنسی پر ایسے تک کہ ہوا جو گئے ہو شارٹ گا	2
		آئندہ جانے کا پروگرام ابھی بنایا نہیں ہے باقی اللہ بہتر جانتا ہے	3
	Bidirectional Encoder Decoder	کل میں نے بڑی مشکل سے انگلش اور تھا والا عنوان آپ کو ٹیگ پھر بھی نہیں ہوا	1
		اس طرح آپ جلد ازکم ہو سکتے ہیں میں کرے ہو فیصلہ سے بڑھانے ممکن کے کر ہونے اندر اپنے گا	2
		آئندہ جانے کا پروگرام ابھی بنایا نہیں ہے باقی کہتا ٹھیک ایکہے	3
	Bidirectional Encoder Decoder with Attention	کل میں نے بڑی مشکل سے انگلش میں تبدیل کیا لیکن آپ کو ٹیگ بھی مجھے نہیں ہوا	1
		اس طرح آپ جلد جزیرے ہو سکتے ہیں اور گھر میں کر کے بیمار ہونے کا خدشہ بھی کم ہو گا	2
		آئندہ جانے کا پروگرام ابھی بنایا نہیں ہے باقی اللہ بہتر جانتا ہے	3
	Original Output	کل میں نے بڑی مشکل سے انگلش میں تبدیل کیا لیکن آپ کو ٹیگ پھر بھی نہیں ہوا	1
		اس طرح آپ جلد صحتیاب ہو سکتے ہیں اور گھر میں دوسروں کے بیما ر ہونے کا خدشہ بھی کم ہو گا	2
		آئندہ جانے کا پروگرام ابھی بنایا نہیں ہے باقی اللہ بہتر جانتا ہے	3

6.2.2. Long Length Sentences

As with short sentences, it can be seen from Table 4 that all the sentences make complete sense and even the wrong predictions are not at all random. Chances of making incorrect predictions are less in sentences that are relatively shorter. Very long sentences are correctly predicted at least half way before the model loses the context of the input sentence when no attention mechanism is used. In this scenario, we can vividly see the effect of using attention mechanism, which keeps the model focused on only the relevant parts and prevents it from getting lost even for very long sentences. We can also see the effect of unknown words using the example of the word صحتیاب. Being an unknown word for the model, the bidirectional encoder with attention chooses the most suitable noun in place of this noun. It can be inferred that the bidirectional encoder with attention aptly overcomes the problems of long sentences and unknown words. However, still further improvement is required.

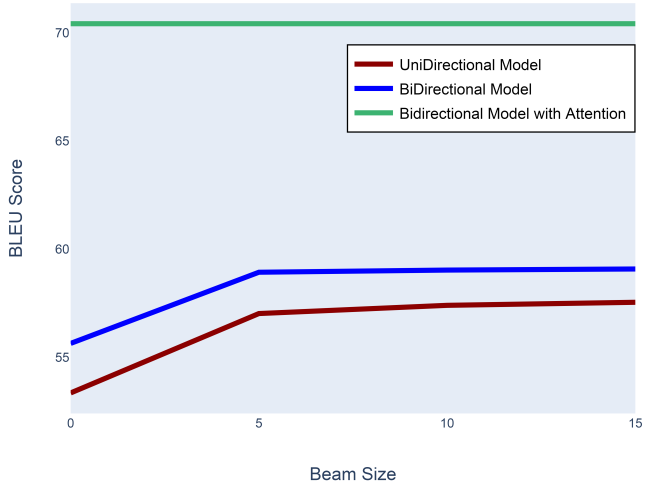

Fig. 6: Effect of beams of sizes 5, 10, and 15 on the BLEU score on all the three models.

6.2.3. Effect of beams

We have empirically evaluated the effect of using beams during the inference task and we tested all the three models without and with beams of sizes 5, 10, and 15. The basic encoder-decoder model and the bidirectional encoder-decoder model

seem to have made a much noticeable jump in the prediction quality when beams are used. However, the bidirectional encoder decoder with attention model did not respond to the beams as shown in Figure 6. This could be attributed to the fact that the model has absorbed all possible variations during training making the use of beams unnecessary at prediction time, at least for the experimented beam sizes. It would be interesting to see the effect (if any) of even larger beams as future work.

6.2.4. Comparison with Dictionary method

It is worth mentioning that dictionary-based one-to-one mapping of Roman-Urdu to Urdu transliteration was only successful for already seen words that did not need any contextual information. However, it failed miserably for unseen words, words with different connotations, and words requiring contextual information. Needless to mention, such mechanism is bound to fail on sentences with unequal word length.

7. Conclusions and Future Work

In this paper, we have given a detailed empirical analysis of three sequence to sequence models on the research problem of Roman-Urdu to Urdu transliteration. To the best of our knowledge, this is the first attempt to address this problem using deep learning techniques. Currently all the existing transliteration models use conventional rule-based or phrase-based statistical models that have limited capacity to generalize, scale, and learn. All three of our models give excellent performance, particularly the model with bidirectional encoder and attention mechanism sets the benchmark of 70 BLEU score. This confirms that the attention mechanism serves as a guideline for the decoder to make better predictions, which is also valid for unknown words. It is also shown empirically that using bidirectional encoder enhances the learning capacity of the model and the use of beams during decoding leads to better predictions. Our models learn all the syntactic, semantic, and contextual information, and give us the translation quality which is very close to human translation. We have also built the first ever Roman-Urdu to Urdu parallel corpus of 1.1 million sentences and made it publicly available.

We plan to extend our work by generating an even bigger dataset and also by observing the effect of residual connections, dropout, making a deeper network, and using other variations of attention mechanism.

Appendix A. English Translations for Urdu and Roman-Urdu

Translation for every Urdu and Roman-Urdu word or sentence in the order of occurrence is given below.

Sr No	Urdu Words	Roman-Urdu Words	English Translation	Page No	Section / Table No
1	عام	aam	common	3	Sec. 1.1

2	آم	aam	mango	3	Sec. 1.1
3	ہم اسکول گئے۔	Hum school gae.	We went to school.	3	Sec. 1.1
4	ہم اسکول گئی۔	Hum school gae.	We (females) went to school.	3	Sec. 1.1
5	یہ اسلام آباد ہے۔	Yeah Islamabad hai.	This is Islamabad.	3	Sec. 1.1
6	کس کیلئے ہیں ؟	Kis ke lye hai?	For whom is this?	3	Sec. 1.1
7	اسلام آباد	Islamabad	Islamabad	11	Sec. 4.4
8	بے وقوف	bewakuf	fool	11	Sec. 4.4
9	علم و ادب	ilmoadam	literature	11	Sec. 4.4
10	سارہ اور زارا دوست ہیں۔	Sara aur Zara dost hain.	Sara and Zara are friends.	11	Table 2
11	بہت شکریہ آپ کا اس عزت افزائی کے لیے۔	bohat shukria aap ka is izzat afzai ke liye.	Many thanks to you for this honour.	15	Table 3
12	بہت خوبصورت تصاویر ہیں۔	bohat khobsorat tasaveer hai.	(They) are very beautiful photos.	15	Table 3
13	غصہ میں کچھ بھی ہو سکتا ہے۔	gussa mein kuch bhi ho sakta hai.	Anything can happen in anger.	15	Table 3
14	آج کے دن تو بہت ہی مصروف رہی ہوں۔	aaj ke din to bohat hi masroof rahi hon.	I have been very busy today.	15	Table 3
15	کچھ احباب کا خیال ہے میں ڈاکٹر ہوں۔	kuch ahbaab ka khayaal hai mein doctor hon.	Some companions think I am a doctor.	15	Table 3
16	عاشورہ کے جلوس پر حملہ ہوا۔	aashorh ke juloos par hamla hua.	The Ashura procession was attacked.	15	Table 3
17	شکریہ احمد صاحب ! یہ نظم میں بھی یہاں پوسٹ کرچکا ہوں۔	Shukria Ahmed sahib! ye nazam mein yahan bhi post kar chukka hon.	Thanks Ahmad sir! I have also posted this poem here.	15	Table 3
18	عاشورہ	aashorh	Ashura (a religious day)	16	Sec. 6.2.1
19	چڑیوں	chiryon	sparrows	16	Sec. 6.2.1
20	بجری	bajri	gravel	16	Sec. 6.2.1
21	چونے	choonay	lime	16	Sec. 6.2.1
22	پوسٹ کرچکا	post ker chukka	has posted	16	Sec. 6.2.1
23	کو اٹھایا	ko uthaya	was picked	16	Sec. 6.2.1
24	کو شامل	ko shamil	was added	16	Sec. 6.2.1
25	کل میں نے بڑی مشکل سے انگلش میں تبدیل کیا لیکن آپ کو ٹیگ پھر بھی نہیں ہوا۔	kal mein ne barri mushkil se english mein tabdeel kya lekin aap ko tag phir bhi nahi hwa.	Yesterday I changed to English with great difficulty but you still did not get tagged.	16	Table 4

26	اس طرح آپ جلد صحتیاب ہو سکتے ہیں اور گھر میں دوسروں کے بیما ر ہونے کا خدشہ بھی کم ہو گا۔	is terhan aap jald shtyab ho satke hain aur ghar mein doosron ke bemaar honay ka khadsha bhi kam ho ga.	This way you can recover quicker and others at home are less likely to get sick.	16	Table 4
26	آئندہ جانے کا پروگرام ابھی بنایا نہیں ہے باقی اللہ بہتر جانتا ہے	aindah jaane ka programme abhi banaya nahi hai baqi Allah behtar jaanta hai.	The program for the future is not yet made, but Allah knows best.	16	Table 4
27	صحتیاب	sehatyaab	healthy	17	Sec. 6.2.2

References

1. P. Agrawal and L. Jain, English to sanskrit transliteration: an effective approach to design natural language translation tool, International Journal of Advanced Research in Computer Science 8(1) (2017) 103–107.
2. M. Alam and S. ul Hussain, Sequence to sequence networks for roman-urdu to urdu transliteration, in Multi-topic Conference (INMIC), 2017 International (2017) pp. 1–7.
3. N. M. Ali, A. El Hamid, M. Mostafa and A. Youssif, Sentiment analysis for movies reviews dataset using deep learning models, Aliaa, Sentiment Analysis for Movies Reviews Dataset Using Deep Learning Models (June 14, 2019) (2019).
4. S. A. Ali, S. Khan, H. Perveen, R. Muzzamil, M. Malik and F. Khalid, Urdu language translator using deep neural network, Indian Journal of Science and Technology 10(40) (2017) 1–7.
5. P. Antony, Machine translation approaches and survey for indian languages, International Journal of Computational Linguistics & Chinese Language Processing 18(1) (2013) 47–78.
6. F. Aqlan, X. Fan, A. Alqwbani and A. Al-Mansoub, Improved arabic–chinese machine translation with linguistic input features, Future Internet 11(1) (2019) 1–17.
7. M. Artetxe, G. Labaka and E. Agirre, An effective approach to unsupervised machine translation, arXiv preprint arXiv:1902.01313 (2019).
8. D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
9. A. V. M. Barone, B. Haddow, U. Germann and R. Sennrich, Regularization techniques for fine-tuning in neural machine translation, arXiv preprint arXiv:1707.09920 (2017).
10. D. Britz, A. Goldie, T. Luong and Q. Le, Massive exploration of neural machine translation architectures, arXiv preprint arXiv:1703.03906 (2017).
11. K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, CoRR abs/1406.1078 (2014).
12. A. Daud, W. Khan and D. Che, Urdu language processing: a sur vey, Artificial Intelligence Review 47(3) (2017) 279–311.
13. M. A. Di Gangi and F. Marcello, Can monolingual embeddings improve neural machine translation?, Proc. of CLiC-it (2017) 141–146.
14. M. Federico, A. Cattelan and M. Trombetti, Measuring user productivity in machine translation enhanced computer assisted translation, in Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA) (2012) pp. 44–56.

15. J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, Convolutional sequence to sequence learning, arXiv preprint arXiv:1705.03122 (2017).
16. A. Graves, Sequence transduction with recurrent neural networks, arXiv preprint arXiv:1211.3711 (2012).
17. S. Green, J. Heer and C. D. Manning, The efficacy of human post-editing for language translation, in Proceedings of the SIGCHI conference on human factors in computing systems (2013) pp. 439–448.
18. R. Grundkiewicz and M. Junczys-Dowmunt, Near human-level performance in grammatical error correction with hybrid machine translation, arXiv preprint arXiv:1804.05945 (2018).
19. M. Hassan and M. Shoaib, Opinion within opinion: Segmentation approach for urdu sentiment analysis, International Arab Journal of Information Technology (IAJIT) 15(1) (2018) 21–28.
20. S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural computation 9(8) (1997) 1735–1780.
21. K. Irie, A. Zeyer, R. Schlüter and H. Ney, Language modeling with deep transformers, arXiv preprint arXiv:1905.04226 (2019).
22. A. H. Jadidinejad, Neural machine transliteration: Preliminary results, CoRR abs/1609.04253 (2016).
23. A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, Bag of tricks for efficient text classification, Computing Research Repository abs/1607.01759 (2016).
24. L. u. Kaiser and S. Bengio, Can active memory replace attention?, in D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett (eds.), Advances in Neural Information Processing Systems 29 (Curran Associates, Inc., 2016) pp. 3781–3789.
25. N. Kalchbrenner and P. Blunsom, Recurrent continuous translation models, in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013) pp. 1700–1709.
26. D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
27. P. Koehn, Europarl: A parallel corpus for statistical machine translation, in MT summit, Vol. 5 (2005) pp. 79–86.
28. G. Lample, M. Ott, A. Conneau, L. Denoyer and M. Ranzato, Phrase-based & neural unsupervised machine translation, arXiv preprint arXiv:1804.07755 (2018).
29. S. Läubli, M. Fishel, G. Massey, M. Ehrensberger-Dow, M. Volk, S. O'Brien, M. Simard and L. Specia, Assessing post-editing efficiency in a realistic translation environment (2013) 83–91.
30. N. T. Le, F. Sadat, L. Menard and D. Dinh, Low-resource machine transliteration using recurrent neural networks, ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 18(2) (2019) 1–14.
31. Q. Le and T. Mikolov, Distributed representations of sentences and documents, in International Conference on Machine Learning (2014) pp. 1188–1196.
32. Y. LeCun, Y. Bengio and G. Hinton, Deep learning, nature 521(7553) (2015) 436–444.
33. S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi and J. A. Benediktsson, Deep learning for hyperspectral image classification: An overview, IEEE Transactions on Geoscience and Remote Sensing (2019) 6690–6709.
34. M. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, Computing Research Repository abs/1508.04025 (2015).
35. M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals and W. Zaremba, Addressing the rare word problem in neural machine translation, arXiv preprint arXiv:1410.8206 (2014).
36. M. M. Mahsuli and R. Safabakhsh, English to persian transliteration using attention-

based approach in deep learning, in Electrical Engineering (ICEE), 2017 Iranian Conference on (2017) pp. 174–178.
37. T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).
38. S. Min, B. Lee and S. Yoon, Deep learning in bioinformatics, Briefings in bioinformatics 18(5) (2017) 851–869.
39. N. Mukhtar and M. A. Khan, Urdu sentiment analysis using supervised machine learning approach, International Journal of Pattern Recognition and Artificial Intelligence 32(02) (2018) 1851001–1–1851001–15.
40. L. Niu and X. Dai, Topic2vec: Learning distributed representations of topics, CoRR abs/1506.08422 (2015).
41. M. Pagliardini, P. Gupta and M. Jaggi, Unsupervised learning of sentence embeddings using compositional n-gram features, CoRR abs/1703.02507 (2017).
42. K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in Proceedings of the 40th annual meeting on association for computational linguistics (2002) pp. 311–318.
43. J. Pastor-Pellicer, M. J. Castro-Bleda, S. España-Boquera and F. Zamora-Martínez, Handwriting recognition by using deep learning to extract meaningful features, AI Communications (Preprint) (2019) 1–12.
44. J. Pennington, R. Socher and C. Manning, Glove: Global vectors for word representation, in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (2014) pp. 1532–1543.
45. G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser and G. Hinton, Regularizing neural networks by penalizing confident output distributions, arXiv preprint arXiv:1701.06548 (2017).
46. W. Ping, K. Peng and J. Chen, Clarinet: Parallel wave generation in end-to-end text-to-speech, arXiv preprint arXiv:1807.07281 (2018).
47. S. Raj, Z. Rehman, S. Rauf, R. Siddique and W. Anwar, An artificial neural network approach for sentence boundary disambiguation in urdu language text, International Arab Journal of Information Technology (IAJIT) 12(4) (2015) 395–400.
48. P. Ramachandran, P. J. Liu and Q. V. Le, Unsupervised pretraining for sequence to sequence learning, arXiv preprint arXiv:1611.02683 (2016).
49. K. Revanuru, K. Turlapaty and S. Rao, Neural machine translation of indian languages, in Proceedings of the 10th Annual ACM India Compute Conference on ZZZ (2017) pp. 11–20.
50. M. Rosca and T. Breuel, Sequence-to-sequence neural network models for transliteration, CoRR abs/1610.09565 (2016).
51. H. Sankar, V. Subramaniyaswamy, V. Vijayakumar, S. Arun Kumar, R. Logesh and A. Umamakeswari, Intelligent sentiment analysis approach using edge computing-based deep learning technique, Software: Practice and Experience (2019) 1–13.
52. M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45(11) (1997) 2673–2681.
53. Z. Sharf and S. U. Rahman, Performing natural language processing on roman urdu datasets, IJCSNS 18(1) (2018) 141–148.
54. N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton and J. Dean, Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, CoRR abs/1701.06538 (2017).
55. R. Shu and A. Miura, Residual stacking of rnns for neural machine translation, in Proceedings of the 3rd Workshop on Asian Translation (WAT2016) (2016) pp. 223–229.

56. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15(1) (2014) 1929–1958.
57. M. Sundermeyer, R. Schlüter and H. Ney, Lstm neural networks for language modeling, in Thirteenth annual conference of the international speech communication association (2012) pp. 194–197.
58. I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks, in Advances in neural information processing systems (2014) pp. 3104–3112.
59. H. Tachibana, K. Uenoyama and S. Aihara, Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention, in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2018) pp. 4784–4788.
60. A. Tezcan, V. Hoste and L. Macken, Estimating word-level quality of statistical machine translation output using monolingual information alone, Natural Language Engineering (2019) 1–22.
61. A. Vassilev, Bowtie-a deep learning feedforward neural network for sentiment analysis, arXiv preprint arXiv:1904.12624 (2019).
62. O. Vinyals and Q. V. Le, A neural conversational model, Computing Research Repository abs/1506.05869 (2015).
63. L. M. Werlen, N. Pappas, D. Ram and A. Popescu-Belis, Self-attentive residual decoder for neural machine translation, arXiv preprint arXiv:1709.04849 (2017).
64. L. M. Werlen, N. Pappas, D. Ram and A. Popescu-Belis, Self-attentive residual decoder for neural machine translation, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Vol. 1 (2018) pp. 1366–1379.
65. Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv:1609.08144 (2016).
66. C. Xiong, S. Merity and R. Socher, Dynamic memory networks for visual and textual question answering, in International Conference on Machine Learning (2016) pp. 2397–2406.
67. B. Zoph, D. Yuret, J. May and K. Knight, Transfer learning for low-resource neural machine translation, arXiv preprint arXiv:1604.02201 (2016).