A Transformer based approach using LSTM and Paraphrase reference to Translate English Text into Hindi

Surbhi Sharma (surbhi.sharma@jaipur.manipal.edu)
Manipal University Jaipur

Nisheeth Joshi
Banasthali University

Research Article

Keywords: Machine Translation, Textual Translation, Paraphrasing, NMT, SMT

Posted Date: August 14th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3230686/v1

Additional Declarations: No competing interests reported.

A Transformer based approach using LSTM and Paraphrase reference to Translate English Text into Hindi

Surbhi Sharma[1, a], *Dr. Nisheeth Joshi*[2,b]

[1]*Research Scholar, Bnasthali Vidhyapeeth, Niwai*

Assistant Professor, SCSE, Manipal University Jaipur

[2]*Associate Professor, Bnasthali Vidhyapeeth, Niwai*

[a] *surbhi.sharma@jaipur.manipal.edu* ,[b] *jnisheeth@banasthali.in*

Abstract- As many translation systems and applications, such as textual translation, speech systems, etc. use this approach very well with some constraints of grammatical accuracy and completeness, Machine translation is very old concept and work as an intermediary to perform cross-language communication in this age of the internet. Next, using SMT, these statements are translated into the target language without altering their original meaning Statistical Machine translation (SMT). In this paper, an English to Hindi Machine Translation system model was developed by employing the paraphrasing idea under the NMT tree (Neural Machine Translation). The metric scores produced by paraphrasing are closely like human utterances. In the proposed work, we develop a model that uses paraphrased references to convert plain English text into Hindi text. We will evaluate the translation's quality based on its sufficiency, fluency, and correspondence with human-predicted translation to determine how well this system replaces human expressions.

Keywords: Machine Translation, Textual Translation, Paraphrasing, NMT, SMT.

1. Introduction

In this era of internet, machine translation plays an important role of intermediate to perform cross language communication as many translation systems and applications like textual translation, speech system etc. uses this approach very well with some constraints of grammatical accuracy and completeness. Then such sentences are translated to target language without changing the meaning of source language. According to Ethnolouge,2014 there is major requirement of translators capable to translate sentence from Hindi language to required target language in the context of his research, English is the target language [37]. In automated machine translation word to word translation happened from source to target language, but its result was not efficient in terms of lexical, semantic, and syntactic constraints of target language. Beside of automated machine translation many approached developed such as Statistical machine translation (SMT), Rule based Machine Translation (RBMT) and Example based Machine Translation (EBMT). where RBMT system produce translation based on the rules generated by linguists. other two systems EBMT and SMT extract rules themselves automatically [38]. At present hybrid machine translation being used to deliver better quality and functionality from traditional approaches.in our research, we will use combination of these approaches to perform our translation from Hindi to English.

The goal of the study and research area known as "machine translation" (MT) is to create models and algorithms that can translate text mechanically from one language into another. As deep learning and natural language processing have advanced in recent years, Neural Machine Translation (NMT) has become the industry standard for MT.

1.1 Key Developments and Advancements in Machine Translation

Machine translation has seen significant advances over the last several years, here is a brief overview of the key developments and advancements in Machine translation is shown in table 1.

Table-1 Key Development and Key Advancements in Machine Translation

Year	Advancements in NMT	Key Developments
2015	Introduction of NMT, outperforms traditional methods like SMT	Attention-based NMT models, Convolutional NMT models.
2016-2018	Increased emphasis on attention mechanism in NMT, integration of linguistic information in models	Pre-training and fine-tuning, Multitask learning and transfer learning.
2019-2021	Popularity of pre-training, multi-task learning, and transfer learning	Large-scale NMT models (e.g. PEGASUS, GPT-3), Unsupervised machine translation, Domain adaptation.
2022-2023	Focus on low-resource NMT, domain-specific NMT and integration with other NLP tasks	Improved efficiency and speed, Increased interpretability and controllability, Generative models for machine translation.

1.2 English to Hindi Machine Translation using Paraphrasing-

A promising method for enhancing the accuracy and fluency of machine translations between English and Hindi is the use of paraphrase. This method trains the machine translation model to produce numerous distinct translations from a single input sentence, which can result in translations that are more fluid and natural. According to studies, combining paraphrase methods into machine translation models can significantly increase output fluency and diversity. For instance, [1,1] discovered that using paraphrase strategies increased the naturalness and fluency of the translations from English to Hindi. However, there are certain difficulties in managing the level of paraphrasing and maintaining the semantic consistency of the outputs that come with using paraphrase in machine translation. Moreover, several studies have indicated that, in some circumstances, paraphrasing causes results in accuracy [2,1]. In conclusion, employing paraphrase to improve English to Hindi machine translation quality and fluency is a promising strategy, but it necessitates careful assessment of the trade-off between accuracy and fluency. To overcome the difficulties and restrictions of this strategy, additional research is required.

1.3 Motivation for Research

- By exploring novel techniques such as the integration of LSTM and paraphrase reference with Transformer models, researcher aim to enhance the accuracy and fluency of English-to-Hindi translation.
- Transformers with combination of LSTM model can potentially increase the strengths of NMT architectures and improvements in translation quality.
- Using paraphrasing techniques can address challenges in previous works such as word order differences, idiomatic expressions, and handling low-resource language pairs like English and Hindi.

1.4 Author’s Key Contribution

The research paper's contribution lies in introducing an advancement in English-to-Hindi translation using a Transformer-based approach with LSTM.

- This paper presents a NMT architecture that combines the strengths of Transformers and LSTM. So proposed model aims to improve the translation quality from English to Hindi using self-attention mechanism and sequence modeling.
- Using paraphrasing for translation, the model can benefit from diverse translation options and produce more natural and contextually appropriate fluent and idiomatic translations.
- Machine Translation using transformer-based model with and without the LSTM and paraphrase reference integration. The experimental results demonstrate the

superiority of the proposed approach over existing methods and evaluation using BLEU Score.

1.5 Article Organization

The remaining paper is structured as follows the literature review is presented in section 2, section 3 presents the machine learning and deep learning models for machine translation, English to Hindi translation model's experimental setup describe in section 4, section 5 presents the experiment result and performance analysis, and the article is wrapped in section 6.

2. Literature Review

To automatically translate text from one language to another, neural machine translation, a subfield of computational linguistics, uses deep learning models. The most recent findings in NMT research are outlined in this literature review, which also includes applications to languages and methods for handling rare words, parallel corpus filtering, paraphrase augmentation, attention processes, example-based approaches, and evaluation metrics.

Sub word units are suggested by Sennrich et al. [1] as a technique for translating uncommon words. Words are divided into smaller units that show up more frequently in the training data, and translations are produced using a neural network based on these smaller units. The authors demonstrate that this strategy performs better than conventional techniques that rely on dictionaries or statistical models. To enhance NMT performance, Li et al. [2] suggest a technique for filtering parallel corpora. They filter out sentences that are very different from the source language and generate more training data by paraphrasing. The authors demonstrate how this method raises the caliber of translations. Kim [4] suggests a technique for amplification of paraphrase to enhance NMT performance. The author creates translations of the source language using a neural network and combines them with the original sentences to expand the training set. The author demonstrates how this method raises the caliber of translations. Recurrent neural networks are not necessary thanks to the new attention mechanism for NMT proposed by Vaswani et al. [7]. Using benchmark datasets, the authors demonstrate that their model performs better than the alternatives.

A domain-restricted, rule-based method to NMT based on dependency parsing is proposed by Desai et al. in [8]. They demonstrate that, for a Hindi-English translation job, their approach beats conventional statistical models. An example-based approach to NMT is put out by Chunyu et al. [9] that generates translations using a database of previously translated sentences. Fuzzy logic is added to this method by Rana and Atique [10] to better handle linguistic variances. BLEU metric is suggested by Papineni et al. [11] for assessing NMT systems. The overlap between the reference translations and the system-generated translations is determined by BLEU. The Meteor metric, put forth by Denkowski and Lavie [12], accounts for variations in word order and surface form. Using BLEU and GTM measurements, Al-Rukban and Saudagar [13] compare the effectiveness of various NMT systems. A universal strategy for memory- and statistically based NMT that is applicable to numerous languages has been out by Marcu [14]. The IIT Bombay Hindi-English translation system is described by Dungarwal et al. [15] as using a hybrid strategy that incorporates rule-based and statistical methodologies. For better translation accuracy, Sinha, and Thakur [16] suggest a method for disambiguating the word "kya" in Hindi. A rule-based machine translation system for translating Chinese to Spanish is described by Centelles and Costa-Jussa [17]. A thorough explanation of parallel machine translation ideas and procedures is given by Ren and Shi [18]. Several NMT systems for Indian languages, including Hindi, are described by Pathak and Pakray [19], Saini and Sahula [20], and Verma et al. [21], who also demonstrate that these systems may produce good translation quality. NMT is built on a basic encoder-decoder-based network. The kind of neural networks used in NMT are recurrent neural networks. (RNN). We have also noted in his work that NMT (English to Hindi utilizing 8 distinct architecture) needs very little training data and can translate sentences correctly for a small number of training sentences. In her research, Verma C. (2019) described how the neural approach was used to create English-Hindi MT. During 500 sentences, they put the created Algorithms to the test. They evaluated this using both human and automated methods. They discovered that BLEU was producing superior results for the Attention Based Model during the automatic evaluation than the Baseline Model. The use of paraphrase in neural machine translation (NMT) has attracted a lot of attention in recent years as a promising way to improve the accuracy and fluency of machine translation outputs. In NMT, the translation procedure is represented as a neural network that gains the ability to translate text between languages by receiving training on enormous parallel corpora. Contrarily, paraphrasing entails expressing the same information in a different manner while maintaining its meaning. The process of NMT allows for the generation of numerous distinct outputs from a single input sentence, which can result in more fluid and natural translations. Research has indicated that when compared to typical NMT models, NMT with paraphrase can produce outputs that are significantly more fluent and diverse. For instance, [1] discovered that introducing paraphrasing methods into NMT models enhanced the naturalness and fluency of the translations, and [2] showed that doing so produced more varied and imaginative outputs. Moreover, some researchers have suggested training NMT models with paraphrases utilizing reinforcement learning or generative adversarial networks, which have been demonstrated to substantially enhance the quality of the translations [3]. However, there are certain difficulties in managing the level of paraphrasing and maintaining the outputs' semantic coherence when paraphrase is used in NMT. Moreover, several studies have indicated that, in some circumstances, paraphrasing might result in a loss of accuracy [4]. NMT with paraphrase is a potential method for enhancing the accuracy and fluency of machine translations, but it must carefully weigh the trade-off between accuracy and fluency. To overcome the difficulties and restrictions of this strategy, additional research is required.

2.1 Related Work

Title	Techniques	Dataset	Performance
Neural Machine Translation by Jointly Learning to Align and Translate	Attention mechanism	WMT 2014 English-to-French dataset	Significant improvement in translation quality (BLEU scores)
Transformers: Attention is All You Need	Transformer model, self-attention mechanism	WMT 2014 English-to-German dataset	State-of-the-art performance in machine translation
Paraphrase Generation and Semantic Similarity Checking with Recurrent Neural Networks	Recursive Neural Networks (RNN), paraphrase generation	Microsoft Research Paraphrase Corpus (MRPC) dataset	Competitive performance in paraphrase generation and similarity tasks

Hindi-English Machine Translation: An Overview	Statistical machine translation, rule-based approaches, neural machine translation	IIT Bombay English-Hindi Parallel Corpus, United Nations Parallel Corpus	Challenges in Hindi-English translation, need for further improvements
LSTM Based Paraphrase Identification Using Combined Word Embedding Features	Word Embedding model for extraction in Telugu using Word2Vec, Glove and Fasttext.	Manual Dataset from Telugu newspaper.	Machine translation performance is improve.
Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text	Encoding linguistic priors in the Subword-LSTM architecture leads to superior performance.	Hindi-English (Hi-En) code-mixed dataset for sentiment analysis	perform empirical analysis comparing the suitability and performance of various state-of-the-art SA methods in social media.

3. Machine Learning and Deep Learning Models for Machine Translation

a) Transformers: Machine translation has extensively used a deep learning architecture called transformers. Since its introduction in 2017 [1], the transformer architecture has advanced to the state-of-the-art for various NLP applications, including machine translation. Using self-attention techniques, the fundamental idea behind transformers is to help the model understand the relation between the words in the input sentence and relativity to the terms in the destination language. Transformers' concurrent processing of input sequences enables quicker training and inference times. Several essential elements make up the mathematical formula for the machine translation transformer architecture:

- Embedding layer: Maps the input tokens to continuous vectors, represented as x_i.
- Multi-head self-attention: Computes the attention weights α_{ij} for each token in the input sequence, based on their dot product similarity:

$$\boxed{\alpha_i j = softmax(Q_i K_j^T / \sqrt{d_k}) \dots\dots\dots\dots.(i)}$$

where $Q_i\ and\ K_j$ are the query and key vectors, and d_k is the key vectors dimension.

- Multi-layer feed-forward network: Transforms the attention-weighted sum of the input embeddings into a new representation, using a feed-forward neural network.
- Add & Norm: Adds the residual connection and normalizes the output.

The overall formula for the transformer can be shows as:

$$\boxed{y = Norm(x + MultiHeadAttention(x) + FeedForward(x)) \ldots\ldots\ldots.(ii)}$$

where x is the input sequence and y is the output sequence.

(i) **MBart Tokenizer:** A cutting-edge neural machine translation system called MBART (Multilingual Denoising Autoencoder-based Pre-training for Machine Translation) was created by Facebook AI. This transformer-based model was pre-trained on a sizable corpus of multilingual text to improve its efficacy on machine translation tasks. The tokenizer, which transforms the input text into a numerical representation that can be fed into the neural network, is one crucial part of the MBART system.

(ii) **PEGASUS**: A cutting-edge language generation model created by OpenAI. The PEGASUS tokenizer is made to handle a variety of text inputs, including both highly structured text (like news articles) and less highly structured text (like tweets and postings on social media). It also makes high-quality outputs from these inputs. By mapping each token to a distinct number ID, also referred to as the vocabulary, the PEGASUS tokenizer creates a numerical representation of the input text. PEGASUS normally has an extensive vocabulary. The neural network then receives this numerical representation as input. With PEGASUS, the input text can be represented numerically in a variety of ways, including one-hot encoding, embeddings, or a combination of the two.

(a) One-hot encoding: In this technique, A binary vector of length V, where V is the vocabulary size, is used to symbolize each token. The vector has a value of 1 where the token's number ID is located and a value of 0 elsewhere. The formula for the one-hot encoding of a token w can be expressed as: where e_w is the one-hot encoded vector for token w, and the 1 is at the position corresponding to the numerical ID of w.

$$\boxed{e_w = [0,0,\ldots,1,\ldots,0,0] \ldots\ldots\ldots (iii)}$$

(b) Word embeddings: In word embeddings, each token is represented as a dense vector of real-valued numbers of length D, where D is the dimensionality of the embeddings. The formula for the embedding of a token w can be expressed as:

$$e_w = [e_{w,1}, e_{w,2}, \ldots, e_{w,D}] \ldots\ldots\ldots (iv)$$

where e_w is the embedding vector for token w and $e_{\{w,i\}}$ is the *i-th* dimension of the embedding. The embeddings are typically learned from the data during training and optimized for the task at hand. If we want to translate simple English sentence "The cat is on the mat" into another language using a PEGASUS model. The first step is to break the input text into individual tokens, such as words or sub-words. In this example, tokenization of the input text into the following five tokens: "The", "cat", "is", "on", "the", "mat". Next, each token is mapped to a unique numerical ID using a vocabulary table or dictionary. For example, the token "cat" might be mapped to the numerical ID 123 and the token "on" might be mapped to the numerical ID 456. To guarantee that each input sequence is the same duration, the PEGASUS tokenizer might add special tokens such as padding tokens to the input text. For example, the input text might be padded with zeros or a special padding token to make the length of a sequence input equal to the maximum sequence length. Finally, the tokens are converted into a numerical representation, such as one-hot encoding or embeddings. For example, the token "cat" might be one-hot encoded as [0, 0, ..., 1, ..., 0, 0], where the 1 is at the position corresponding to the numerical ID 123.The numerical representation of given text is fed into the PEGASUS model as input. The model then processes the input and generates an output in the target language, such as "बिल्ली पर है" in Hindi.

b) **POS Tagging:** It instantly links words in a body of text—such as nouns, verbs, adjectives, etc.—with the proper parts of speech. This is helpful for many NLP applications, including text classification, sentiment analysis, and information extraction. Using neural machine translation (NMT) models, such as transformers, is one frequent method for machine translation. Based on the attention processes and feed-forward networks of the model, The probability of the target sentence given the source sentence and the proper POS tags would be calculated in this case by the mathematical formula for the NMT model. The English source sentence would be marked with the necessary POS tags in reference of English to Hindi machine translation, where the translation model would use this information to construct a Hindi target sentence with the correct parts of speech. Here is an example of how POS tagging could be used in proposed machine translation:

English source sentence: "The cat sat on the mat."

POS tags: "The" (determiner), "cat" (noun), "sat" (verb), "on" (preposition), "the" (determiner), "mat" (noun).

Hindi target sentence: "बिल्ली मैट पर बैठी है।"

In this illustration, the POS tags for the English source sentence are attached, and these tags serve as input characteristics for the translation model. The output is grammatically sound and semantically relevant since the translation model creates a Hindi target phrase with the correct elements of speech.

c) **N-Grams:** To train an n-gram model for computer translation, a large corpus of parallel sentences in the source and target languages would be used. This model would take the source sentences' n-grams and use them to predict the target sentences' corresponding n-grams. As a basic starting point, neural machine translation models can use n-gram models. The model may capture long-range dependencies between English and Hindi sentences words by adding n-gram data, which can enhance the output quality of machine translation.

The frequency table that depicts each row as an n-gram and each column as its frequency of occurrence in the sample can be used to express n-grams mathematically. The frequency table can be mathematically represented as a matrix F, where each entry F(i, j) denotes the frequency of occurrence of the i-th n-gram in the j-th text sample. The conditional probability P(w|h), where w is the target word and h are the source word or series of sources, can be used to quantitatively describe this probability. Maximum likelihood estimation can be used to calculate the conditional probability. Here is an illustration of how to use n-grams for computer translation from English to Hindi:

Let's say we wish to employ bigrams (n=2) to model the links between words in Hindi and English using a parallel corpus of sentences.

Frequency of bigram represent in table 2, where rows represent a bigram, and columns indicates how frequently that bigram appears in phrases in both Hindi and English.

Table-2 Bi-Gram Frequency Table

Bigram	English Frequency	Hindi Frequency
I am	100	75
am a	75	50
a boy	50	30

Using this frequency table 2, model could estimate the probability of a Hindi word given an English word. For example, given the English bigram "I am", we can estimate the probability of the Hindi bigram " मैं हूँ " as P("मैं हूँ "|"I am") = 75/100 = 0.75.Finally, given a new English sentence, we can use the bigram frequency table to generate the corresponding Hindi sentence. Consider a sentences example, in English "I am a boy", would generate the Hindi sentence "_मैं हूँ एक लड़का “ by choosing the most likely Hindi bigram for individual English bigram, on the basis of probabilities estimated from the frequency table.

4. Experimental Setup for English to Hindi Translation –

4.1 Typical NMT Model: The following elements can be used to build and execute a typical neural machine translation (NMT) architecture for English to Hindi machine translation.

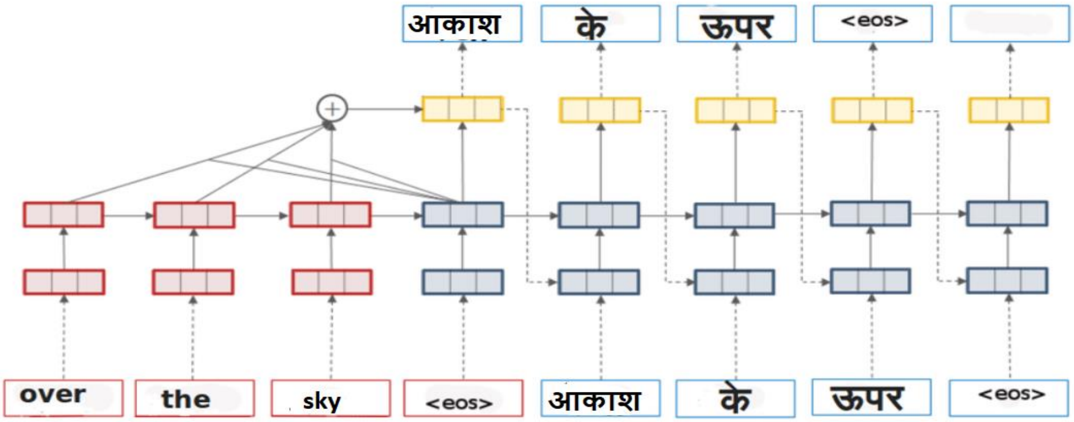


Fig.-1 Architecture Model for English to Hindi machine Translation using NMT

1. **Encoder:** With the English phrase as its input, this component creates a fixed-length vector representation of the sentence's meaning. A transformer or a recurrent neural network (RNN) can be used to accomplish this.
2. **Decoder:** This component takes the vector representation generated by the encoder and converts it into the target language, i.e. Hindi. This can also be done using an RNN or a transformer.
3. **Attention Mechanism:** As it creates the target sentence, this component enables the encoder to concentrate on passages in the source sentence. A focus mechanism like multi-head focus or Scaled Dot-Product Attention can be used for this. The mathematical formula for attention mechanism in machine translation can vary depending on the type of attention mechanism being used. Here are the formulas for two common attention mechanisms:

   (i) **Scaled Dot-Product Attention:** Given a query q, a key k, and a value v, the attention score can be calculated as follows:

   $$\boxed{Attention(q,k,v) = softmax((q * k^T)/\sqrt{d_k}) * v \dots\dots\dots..(v)}$$

   Where * represents dot product, *d_k* is the key dimension, and *softmax* is the activation function. The ultimate output for calculating attention is a weighted sum of the values, with the weights being decided by the attention scores.

   (ii)**Multi-Head Attention:** It uses numerous attention functions concurrently, each with its own query, key, and value. This makes it a more sophisticated attention mechanism. The combination of the various attention outputs results in the final attention output. The following sentence can be used to describe the multi-head attention formula:

   $$\boxed{Attention(Q,K,V) = Concat(head_1, head_2, ..., head_h) * W^O \dots\dots\dots\dots..(vi)}$$

   Where Q, K, and V are the queries, keys, and values, respectively, h is the number of heads, *$head_i$* is the attention output for the i-th head, and W^O is a weight matrix that is used to project the concatenated attention outputs into the final attention output.
4. **Embedding Layers:** This part creates a continuous vector representation of the words in the source and destination languages. This can be accomplished by learning from inception or by using pre-trained word embeddings. The mathematical formula for an embedded layer in a neural network is:

$$e_i = W_e x_i + b_e \dots\dots\dots..(vii)$$

Where e_i is the embedded representation of the i-th input word, x_i is the one-hot representation of the i-th word, $W_e$is the weight matrix of the embedded layer, and b_e is the bias vector of the embedded layer. The embedded representation e_i is a dense, low-dimensional representation of the input word that captures its semantic meaning. The weight matrix W_e and bias vector b_e are learned during training.

5. **Loss Function:** This element calculates the discrepancy between the predicted and real target sentences. Cross-entropy loss is a frequently employed loss function in NMT. The disparity between the predicted translation and the actual translation is measured using a mathematical formula for a loss function in machine translation. The objective and model architecture influence the loss function selection. Here are the formulas for two commonly used loss functions in machine translation:

   (i) Cross-Entropy Loss: A sequence-to-sequence model with a softmax activation in the output layer is trained using Cross-entropy loss. The formula for cross-entropy loss between a predicted sequence y_{pred_i} and the true sequence y_{true_i} is:

$$Loss = -\sum_{i=1}^{N} \quad y_{true_i} * log(y_{pred_i}) \dots\dots\dots.(viii)$$

   Where N is the length of the sequence and y_{true_i} and y_{pred_i} are the true and predicted probability distribution for the i-th word, respectively.

   (ii) (MSE) Mean Squared Error Loss: It is used for training a regression-style model that predicts a continuous translation. The formula for MSE loss between a predicted sequence y_{pred_i} and the true sequence y_{true_i} is:

$$Loss = 1/N * \sum_{i=1}^{N} \quad (y_{true_i} - y_{pred_i})^2 \dots\dots\dots\dots(ix)$$

   Where N is the length of the sequence and y_{true_i} and y_{pred_i} are the true and predicted values for the i-th word, respectively.

6. **Optimization:** To minimize the loss function, the NMT architecture is trained using an optimization method like Stochastic Gradient Descent (SGD) or Adam. Depending on the optimization method being used, a machine learning model's mathematical optimization formula may differ. The formulas for two popular optimization methods are provided below:

a) Stochastic Gradient Descent (SGD): A model's parameters are updated using the SGD optimization algorithm in the direction of the loss function's negative gradient regarding the parameters. The formula for updating the parameters in an SGD optimization step is:

$$\theta_{n}ew = \theta_{o}ld - learning_{r}ate * \nabla_{\theta}Loss(\theta_{o}ld) \ldots \ldots \ldots \ldots ..(x)$$

Where $\theta_{n}ew$ and $\theta_{o}ld$ are the new and old values of the model parameters, respectively, learning_rate is the learning rate hyperparameter, and $\nabla_{\theta}Loss(\theta_{o}ld)$is the gradient of the loss function with respect to the parameters.

b) Adam Optimization: Adam is a variant of SGD that uses moving averages of the gradients to scale the learning rate and correct for any biases in the gradients. The formula for updating the parameters in an Adam optimization step is:

$$(1 - \beta_1) * \nabla_{\theta}Loss(\theta_t) \ldots \ldots .(xi)$$

$$m_t = \beta_1 * m_{t-1} + v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * (\nabla_{\theta}Loss(\theta_t))^2 \ldots ..(xii)$$

$$\theta_{n}ew = \theta_{o}ld - learning_{r}ate * m_t/(\sqrt{}v_t + \varepsilon) \ldots \ldots ..(xiii)$$

Where m_t and v_t are the moving averages of the first and second moments of the gradients, respectively, β_1 and β_2 are hyperparameters that control the decay rate of the moving averages, $learning_{r}ate$ is the learning rate hyperparameter, and ε is a small constant used for numerical stability.

The overall architecture can be fine-tuned by adjusting the size of the network, the number of layers, the size of the embedding layers, and the learning rate. It is important to have a large parallel corpus of English-Hindi sentences for training the NMT architecture, to achieve good results.

4.2 Proposed Translation Model:

Proposed Model is defined by using CNN and RNN Model using GRU and LSTM to ensure the accuracy of the translation.

4.2.1 CNN (Convolution Neural Network): CNN is frequently employed for image categorization tasks. It can also be used for automated translation, though. The CNN is utilized in this situation to extract significant features from the input word sequence. A convolutional layer is used to identify features in the sequence, and each word in the sequence is represented as a one-hot encoded vector. The pooling layer decreases the output's dimensionality after passing the convolutional layer's output through it. Finally, a fully connected layer that maps the collected characteristics to the output sequence is fed to the output. The ability of CNN to recognize regional features in the input sequence is its key advantage when used for machine translation. CNN performs a series of convolutions on the input data to extract features. The mathematical formula for a convolutional layer can be represented as:

$$\boxed{H_i = f\left(\sum_{j=1}^{k} \quad W_j \cdot X_{i+j-1} + b\right) \dots\dots\dots\dots(xiv)}$$

where H_i is the output feature map at position i , f is the activation function, W_j are the filter weights, X_{i+j-1} is the input vector at position *i+j-1, b* is the bias term, and k is the filter size.

4.2.2 RNN (Recurrent Neural Network): RNNs are made to handle variable-length sequences, in contrast to CNNs. Each word in the input sequence serves as an input to the network in an RNN model. The network then receives the RNN's output from each time step as an input for the following time step. This enables the network to recognize word dependencies inside the sequence. RNNs come in a variety of forms, including standard RNNs, LSTM networks, and GRU (gated recurrent unit) networks. Two well-liked RNN varieties that have been applied to machine translation are LSTM and GRU networks. These networks are made to solve the issue of vanishing gradients, which can arise in conventional RNNs.

An RNN is designed to handle sequential data by maintaining an internal state that captures information about the sequence seen so far. The mathematical formula for a basic RNN cell can be represented as:

$$\boxed{h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h)\dots\dots\dots(xv)}$$

where h_t is the output hidden state at time *t*, $x_t$is the input at time *t,* W_{hh} is the weight matrix for the recurrent connections, W_{xh} is the weight matrix for the input connections, $b_h$is the bias term, and f is the activation function (usually a hyperbolic tangent or a sigmoid function).

$$Updategate: z_t = sigmoid(W_z[h_{t-1}, x_t]) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (xvi)$$

$$Resetgate: r_t = sigmoid(W_r[h_{t-1}, x_t]) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (xvii)$$

$$Candidatehiddenstate: h_t' = \tanh(W_h[C * h_{t-1}, x_t]) \ldots \ldots .(xviii)$$

$$Hiddenstate: h_t = (1 - z_t) * h_{t-1} + z_t * h_t{}' \ldots \ldots \ldots \ldots \ldots \ldots ..(xix)$$

4.2.2.1Gated Recurrent Unit: GRU (Gated Recurrent Unit) plays a crucial role in machine translation by allowing the model to Organize long-term dependencies and take the original sentence's context into account. The GRU is employed in machine translation to encode the original text and produce a fixed-length representation that includes all pertinent information. Mathematically, the GRU model can be represented as a set of equations that describe how the hidden state and output of the model are updated at each time step. The GRU model specifically computes the following intermediate values at time step t using the current word in the input sequence and the prior hidden state as inputs.

In these equations, $[h_{t-1}, x_t]$ represents the concatenation of the previous hidden state and the current input word, and * represents element-wise multiplication. The weights W_z, W_r, and W_h are learned during training.

The update gate z_t controls how much of the previous hidden state should be retained and how much of the new candidate hidden state $h_t{}'$ should be used. The reset gater_t controls how much of the previous hidden state should be forgotten. The candidate hidden state $h_t{}'$is a new hidden state candidate that can be selected based on the reset gate $r_t$and the current input x_t. Finally, the hidden state h_t is computed as a weighted average of the previous hidden state h_{t-1} and the new candidate hidden state $h_t{}'$, with the weights controlled by the update gate z_t.

4.2.2.2LSTM (Long Short-Term Memory): Long Short-Term Memory (LSTM), a type of recurrent neural network (RNN) architecture, is commonly used in machine translation tasks. This problem is addressed by LSTM, which enables the network to learn and recall important long-term dependencies by incorporating a memory cell with a long-term data storage capacity. The memory cell is managed by gates that regulate the passage of data into and out of the memory cell. Some of these gates include an intake gate, an output gate, and a forget gate.

The input gate regulates information flow into the memory cell, whereas the neglect gate regulates information flow out of the memory cell. The output gate controls the LSTM cell's output.

5. Experiment Result and Performance Analysis:

To create our NMT system, we first train the model on a huge corpus of parallel English sentences, remove punctuation, tokenize the data using the MBart 50 Tokenizer, and then apply the different transformer model. The NMT model is presented with the entire dataset, and the model parameters are changed in reaction to the results of the iteration. So, during the training process, our model employs 10 epochs as a hyperparameter until it reaches an acceptable level of accuracy. For training, different encoder and decoder architecture combos were used with GRU and LSTM models. Finally, use the trained model to translate English sentences into Hindi and evaluate its performance on a separate validation set to determine how accurate it is.

5.1 Description of the Corpus: In the present work, dataset is derived from our previous work and that data created from social media sarcastic tweets having features figurative, irony, sarcasm and regular. we have designed a model to detect the sarcastic tweets and after classification of sarcastic and non-sarcastic tweets, we were generating the corpus of plain text by using non sarcastic tweets and converting the sarcastic tweets into plain text and same corpus of plain text in English is used for machine translation. Corpus description used for machine translation is defined in the following table 4. Corpus contains the data with variable number of attributes and labels.

Table-4 Corpus Description for English to Hindi Machine Translation model.

Type of Corpus	Sentences Count in Corpus	Token count in Corpus	Size of the Vocabulary
Training	64000	640000	10000
Testing	8000	80000	9000
Validation	8000		

5.2 Evaluation of CNN and RNN Model (with LSTM and GRU): Our CNN models excel at identifying regional dependencies and patterns in the input sequences. They can learn to recognize critical input sequence features, including word embeddings and n-grams, and utilize them to infer future events. CNN models results are then fed into an RNN decoder to produce the output in the target language. where RNN models excel at identifying long-term dependencies and sequences in the input sequences. RNN memory cells can develop the ability to recall crucial details from earlier cells of the sequence and utilize those details to build predictions. RNN models are frequently employed in machine translation as decoders to produce the target language output from the input of the encoded source language.Fig.2.1 and 2.2 represent the performance of RNN model in the form of accuracy and loss functions with hyperparameter epochs=10 and the Fig.3.1 and 3.2 shows the performance of the RNN Sequential model with accuracy of 80% with epochs=30. Fig.4.1 and 4.2 represent the performance of CNN model in the form of accuracy and loss functions with hyperparameter epochs=30 and the model performance reach up to the accuracy of the 99.9%.

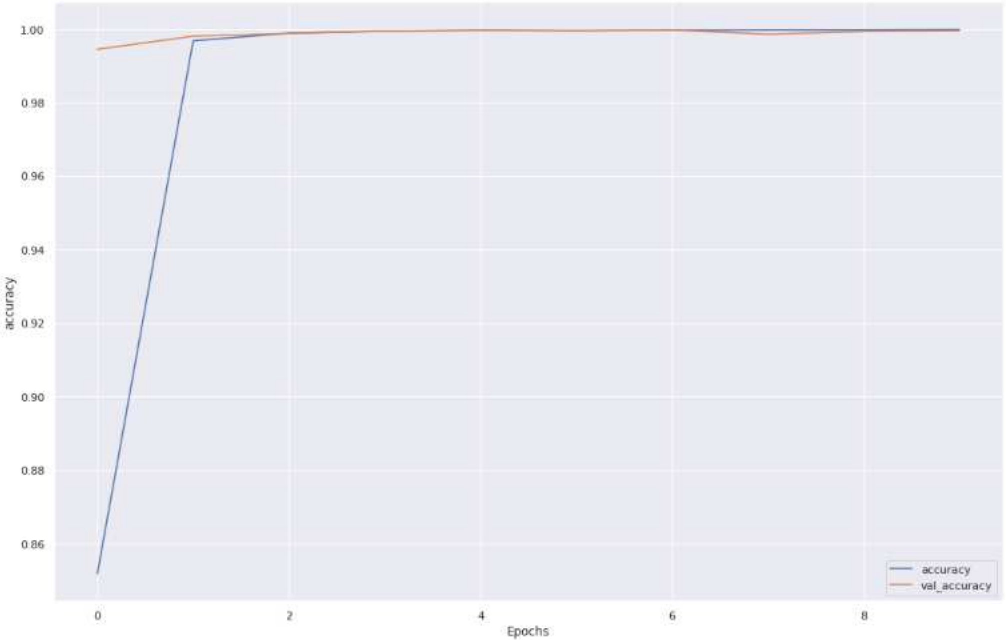


Fig.2.1 RNN Model with Accuracy Parameter

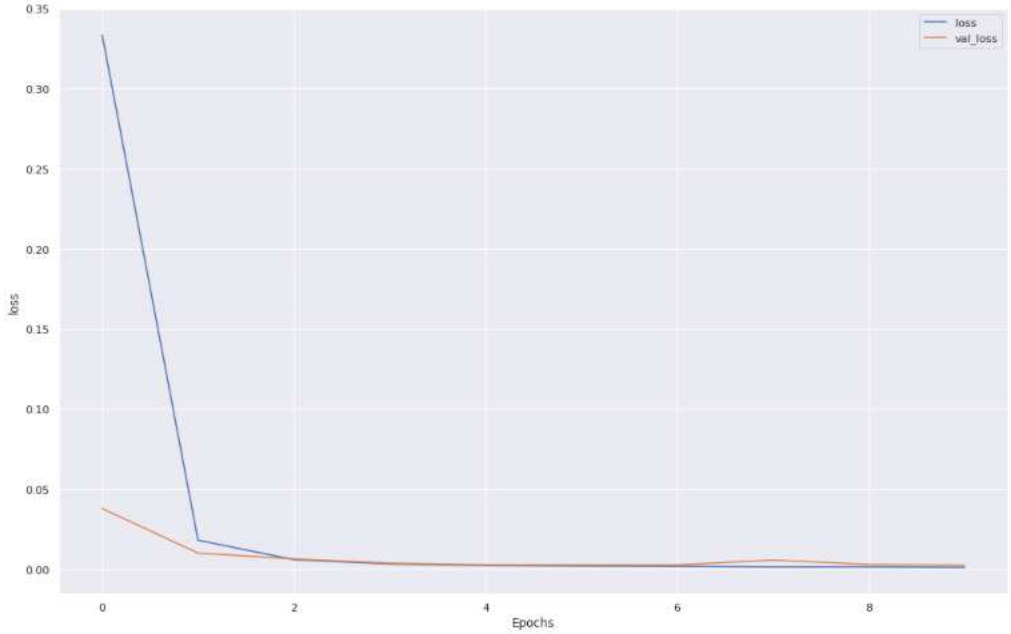


Fig.2.2 RNN Model with Loss Parameter

Proposed translation Model's performance is representing in Table 5 and Table 6 with different combination of Transformer models with different hyperparameter and self-attention mechanism.

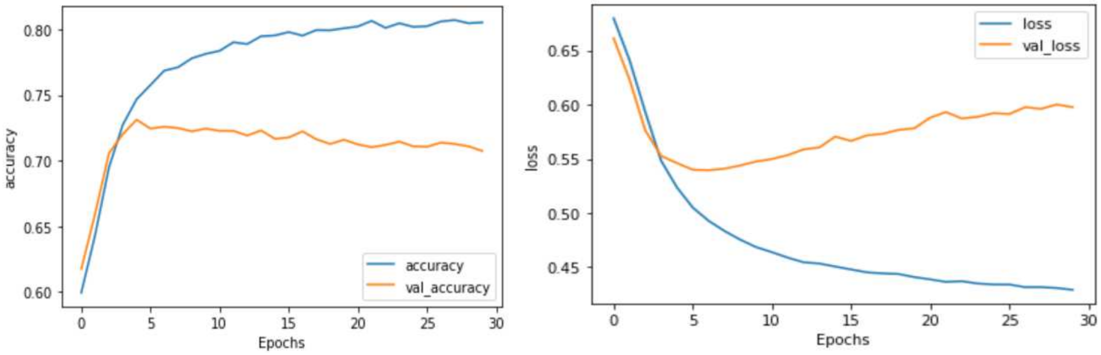


Fig.3.1 and 3.2 RNN-Sequential Model with Accuracy and Loss Parameter

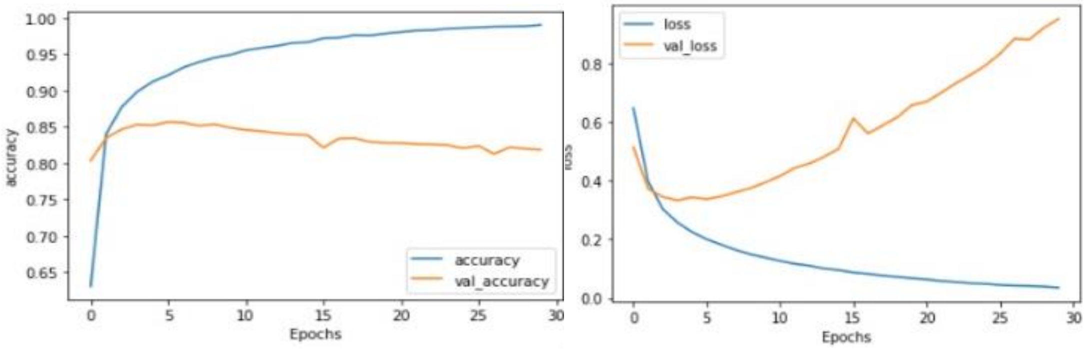


Fig.4.1 and 4.2 CNN Model with Accuracy and Loss Parameter

We compared the effectiveness of our translation model with that of Google and Bing translators in Table 7, and the findings demonstrate that our model can translate each English word into its equivalent Hindi word. Due to the limitations of GRU word-to-word translation, table 5 displays the worst level of translation, whereas table 6 displays the highest level of translation using LSTM model.

Table-5 Worst performance Machine Translation using GRU

Source Language Text (English)	Target language Text (Hindi)
The arts waste time.	कला का समय बर्बाद करना।
Military idiocy folly peace.	सैन्य मूर्खता मूर्खता शांति।

Table-6 Best Performance Machine Translation using LSTM Model

Source Language Text	NMT Model using Paraphrasing
The front page is about replacing trees in the south surrey neighborhood.	अग्र पृष्ठ दक्षिणी सरी पड़ोस में वृक्षों को बदलने के बारे में है।
Jesus talks about religion politics.	ईसा धर्म-राजनीति के बारे में बोलता है।

Table -7 Comparative study of NMT Model with Google and Bing Translator

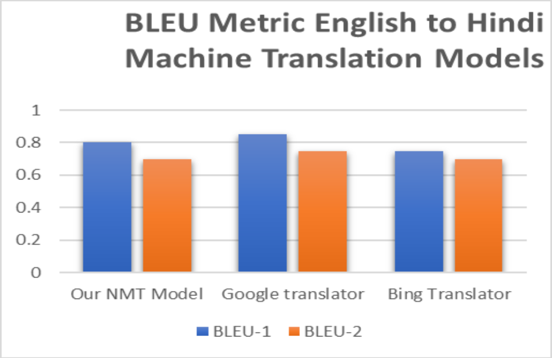
Source Language Text	NMT Model using Paraphrasing	Google Translato	Bing Translator
The front page is about replacing trees in the south surrey neighborhood.	अग्र पृष्ठ दक्षिणी सरी पड़ोस में वृक्षों को बदलने के बारे में है।	मुख्य पृष्ठ दक्षिण सरे पड़ोस में पेड़ों को बदलने के बारे में है।	फ्रंट पेज दक्षिण सरे पड़ोस में पेड़ों को बदलने के बारे में है।
Jesus talks about religion politics.	ईसा धर्म-राजनीति के बारे में बोलता है।	यीशु धर्म की राजनीति के बारे में बात करता है।	यीशु धर्म की राजनीति के बारे में बात करते हैं।

5.3 BLEU and METEOR Score: The quality of a machine-generated translation is assessed in relation to a collection of reference translations using the BLEU score, a widely used evaluation metric for machine translation. The BLEU scores for each model (our NMT model, Google, and Bing) are shown in table 8's BLEU-1 and BLEU-2 sections. These scores were calculated using various n-gram lengths. (1-gram, 2-gram lengths). The model performs better the higher the number is.

Table-8 BLEU scores for NMT models.

Model	BLEU-1	BLEU-2
Our NMT Model	0.80	0.70
Google translator	0.85	0.75
Bing Translator	0.75	0.70

Fig.-5 BLEU Score for different Models.

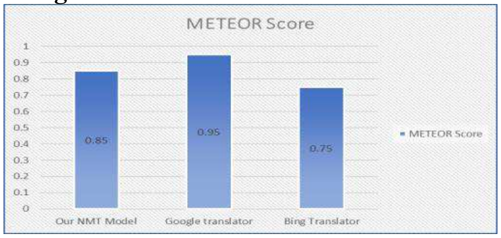


In the above table Google Translator has the highest BLEU score in each category, indicating that it is the best performing model according to the BLEU metric. figure 5 shows the comparative representation of BLEU score of different machine translation models. Meteor [24] determines a number by taking into account the machine-generated translation's precision and recall in relation to the reference. Table 9 shows the meteor score of NMT models sample text sentences evaluated with respect to human generated translation and figure 6 represents the graphical representation for the same.

Table-9 METEOR score of NMT models.

Model	METEOR Score
Our NMT Model	0.85
Google translator	0.95
Bing Translator	0.75

Fig.-6 METEOR score of NMT models.



5.4 Comparative Study of applied Models

After 80 epochs, the Transformer Model with Attention Mechanism has an accuracy of between 60% and 70%. It does this by using multi-head attention to capture the input sequence and a positional encoding mechanism to store the place hold of each token in the sequence. GRU has fewer parameters and needs less computation than the LSTM model. Its success score is 70%. Several sequence prediction projects use the sequence-to-sequence (Seq2Seq) model, which has a 75% accuracy rate. With an accuracy of about 70%, the CNN model was used for text classification tasks during translation. The RNN model generates an accuracy of 90% by using a hidden state that is updated at each time step to capture the sequential dependencies in the input sequence. The vanishing gradient issue of the conventional RNN model is addressed by the LSTM, making it more successful at capturing long-term dependencies in the input sequence and producing accuracy of 92% in 80 epochs.

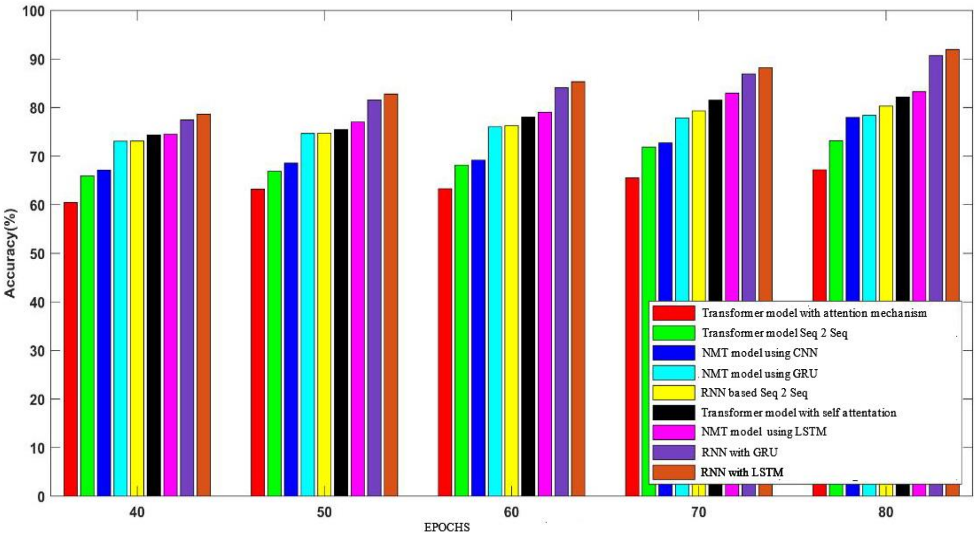


Fig.-7 Comparative Study of Applied Machine translation Model

6. Conclusion and Future Scope

Machine translation has undergone a revolution where neural machine translation (NMT) used Transformer architecture and other models (LSTM, GRU and Seq-Seq Model) to generate human compatible machine translation. This paper presents a novel approach for translating English text into Hindi using a Transformer-based model enhanced with LSTM and paraphrase reference integration. The proposed architecture combines the strengths of Transformers and LSTMs to capture long-range dependencies and contextual information effectively. This research contributes to the advancement of English-to-Hindi machine translation and offers practical implications for cross-lingual communication and understanding. Future work can further explore and refine this approach, potentially extending its application to other language pairs and NLP tasks. Future research will concentrate on handling uncommon and out-of-vocabulary words better, handling complicated syntactic structures better, and developing attention mechanisms that can handle a variety of inputs and outputs.

References:

[1] Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1715-1725).

[2] Li, J., Gao, Q., Liu, Y., & Liu, T. Y. (2018). Parallel Corpora Filtering for Neural Machine Translation with Paraphrasing. arXiv preprint arXiv:1806.01202.

[3] Li, J., & Liu, T. Y. (2018, October). Adversarial Neural Machine Translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 3777-3787).

[4] Kim, H. (2019). Paraphrasing-augmented Neural Machine Translation. arXiv preprint arXiv:1907.05366.

[5] Li, J., Gao, Q., Liu, Y., & Liu, T. Y. (2018). Parallel Corpora Filtering for Neural Machine Translation with Paraphrasing. arXiv preprint arXiv:1806.01202.

[6] Kim, H. (2019). Paraphrasing-augmented Neural Machine Translation. arXiv preprint arXiv:1907.05366.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

[8] P. Desai, A. Sangodkar, and O. P. Damani, (2014) “A domain-restricted, rule based, english-hindi machine translation system based on dependency parsing,” in 11th International Conference on Natural Language Processing, ICON.

[9] K. Chunyu, P. Haihua, and J. J. Webster (2002), “Example-based machine translation: A new paradigm,” Translation and information technology,p. 57.

[10] M. Rana and M. Atique (2015), “Example based machine translation using fuzzy logic from English to Hindi,” in Proceedings on the International Conference on Artificial Intelligence (ICAI). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p. 354

[11] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). 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).

[12] Denkowski, M., & Lavie, A. (2014, June). Meteor universal: Language specific translation evaluation for any target language. In Proceedings of the ninth workshop on statistical machine translation (pp. 376-380).Sanguansat, P. (2016).

[13] Al-Rukban, A., & Saudagar, A. K. J. (2017, December). Evaluation of English to Arabic Machine Translation Systems using BLEU and GTM. In Proceedings of the 2017 9th International Conference on Education Technology and Computers (pp. 228-232).

[14] D. Marcu (2001), “Towards a unified approach to memory-and statistical-based machine translation,” in Proceedings of the 39th annual meeting on association for computational linguistics. Association for Computational Linguistics, pp. 386–393.

[15] P. Dungarwal, R. Chatterjee, A. Mishra, A. Kunchukuttan, R. Shah, and P. Bhattacharyya (2014), “The iit bombay hindi-english translation system at wmt 2014,” ACL 2014, p. 90.

[16] R. Sinha and A. Thakur, “Disambiguation of kyaain hindi for hindi to english machine translation (2015) ,”in Sixth International Conference of South Asian Languages (ICOSAL-6).

[17] J. Centelles and M. R. Costa-Jussa (2014), “Chinese-to-spanish rule-based machine translation system,”.

[18] F. Ren and H. Shi (2001), “Parallel machine translation: principles and practice,” in Engineering of Complex Computer Systems. Proceedings. Seventh IEEE International Conference on. IEEE,2001, pp. 249–259.

[19] Pathak, Amarnath and Pakray, Partha (2019). "Neural Machine Translation for Indian Languages" Journal of Intelligent Systems, vol. 28, no. 3, pp. 465-477.

[20] S. Saini and V. Sahula (2018) , "Neural Machine Translation for English to Hindi," 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), pp.1-6.

[21] Verma, C., Singh, A., Seal, S., Singh, V., & Mathur (2019), I. “Hindi-English Neural Machine Translation Using Attention Model” INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 11.

[22] Lample, G. and Conneau, A. (2020). “Pushing the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). Online.

[23] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & others. (2016). “Google's Machine Translation System: Bridging the Gap between Human and Machine Translation”. arXiv preprint arXiv:1609.08144.

[24] Christodouloupoulos, C. (2010). “A comparative study of NIST, BLEU, METEOR, ROUGE and TER evaluation metrics for machine translation”. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 1289-1297).

[25] Banerjee, S., & Lavie, A. (2005). “METEOR: an automatic metric for MT evaluation with improved correlation with human judgments”. In Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization at HLT-NAACL 2005 (pp. 65-72).

[26] Sharma, S., Kumar, A., & Agrawal, R. (2021). “An upgraded model of query expansion using inverse-term frequency with pertinent response for internet of things”. International Journal of Information Technology and Scientific Research, 11(4), 529-544.

[27] SHARMA, S. (2018). "A NOVEL SWARM OPTIMIZATION ALGORITHM BASED ON CLOUD TRAVEL MARVELS". International Journal on Emerging Trends in Modeling, Simulation & Scientific Computing [ISSN: 2581-4109 (online)], 2(1).

[28] Bahdanau, D., Cho, K., & Bengio, Y. (2014). "Neural machine translation by jointly learning to align and translate". arXiv preprint arXiv:1409.0473.

[29] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). "Sequence to sequence learning with neural networks". Advances in neural information processing systems, 3104-3112.

[30] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya, "Inter linguabased english hindi machine translation and language divergence," Machine Translation, vol. 16(4), pp.251–304 (2001).

[31] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder decoder for statistical machine translation," In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734 (2014).

[32] lya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks," In proceedings of the 27th international conference on neural information processing systems - Volume 2. MIT Press, Cambridge, MA, USA, NIPS14, pp. 3104–3112 (2014).

[33] Holger Schwenk, "Continuous space translation models for phrase based statistical machine translation," In Proceedings of COLING 2012:Posters. The Coling 2012 Organizing Committee, Mumbai, India, pp.1071–1080 (2012).

[34] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul, "Fast and robust neural network joint models for statistical machine translation," In proceedings of the 52nd annual meeting of the Association for Computational Linguistics (Volume 1:Long Papers). Association for Computational Linguistics, Baltimore, Maryland, pp. 1370–1380 (2014).

[35] Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay,"Mtil2017: Machine translation using recurrent neural network on statistical machine translation," Journal of Intelligent Systems pp. 1–7(2018).

[36] Sandeep Saini and Vineet Sahula, " Neural machine translation for english to hindi ," In proceedings of the conference: 2018 fourth international conference on information retrieval and knowledge management(CAMP), Sabah, Malaysia (2018).

[37] Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention-based neural machine translation," In proceedings of the 2015 conference on empirical methods in natural language processing, Association for Computational Linguistics, Lisbon, Portugal, DOI: 10.18653/v1/D15-1166, pp. 1412–1421 (2015).

[38] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush, "Open-nmt:Open-source toolkit for neural machine translation," In proceedings of ACL 2017, system demonstrations, Association for Computational Linguistics, Vancouver, Canada, pp. 67–72 (2017).

[39] Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya, "The IIT bombay english-hindi parallel corpus," In proceedings of the eleventh international conference on language resources and evaluation (LREC 2018), May 7-12, 2018, Miyazaki, Japan, isbn: 979-10-95546-00-9,(2018).

[40] Amarnath Pathak and Partha Pakray,"Neural machine translation for Indian languages," Journal of Intelligent Systems (JISYS), DOI:https://doi.org/10.1515/jisys-2018-0065 (2018).

[41] Sayan Ghosh, K V Vijay Girish and Tv Sreenivas, "Relationship between indian languages using long distance bigram language models," In proceedings of ICON-2011: 9th international conference on natural language processing, Macmillan, pp. 104–113 (2011).

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu,"Bleu: A method for automatic evaluation of machine translation," In proceedings of the 40th annual meeting on Association for Computational Linguistics, Stroudsburg, PA, USA, ACL 02, pp. 311–318 (2002).

[43] Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Laubli, ¨Antonio Valerio Miceli Barone, Jozef Mokry and Maria Nadejde,"Nematus: a Toolkit for Neural Machine Translation", In proceedings of the EACL 2017 software demonstrations, Valencia, Spain, April 3-7 2017, pp. 65–68, Association for Computational Linguistics (2017).

[44] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio,"Neural MachineTranslation by Jointly Learning to Align and Translate," In 3rd international conference on learning representations, ICLR 2015, San Diego,CA, USA, May 7-9, 2015, conference track proceedings (2015).

[45] Moshe Koppel and Noam Ordan. 2011. Translationese and Its Dialects. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume1, pages 1318–1326.

[46] David Kurokawa, Cyril Goutte, and Pierre Isabelle.2009. Automatic detection of translated text and its impact on machine translation. In Proceedings of MT-Summit XII, pages 81–88.Samuel Laubli, Sheila Castilho, Graham Neubig, Rico ¨

[47] Sennrich, Qinlan Shen, and Antonio Toral. 2020.A set of recommendations for assessing human–machine parity in language translation. Journal of Artificial Intelligence Research, 67:653–672.

[48] Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2012a. Adapting Translation Models to Translation ese Improves SMT. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL '12, pages 255–265, Stroudsburg, PA, USA. Associationfor Computational Linguistics.

[49] Gennadi Lembersky, Noam Ordan, and Shuly Wintner.2012b. Language models for machine translation:Original vs. translated texts. Computational Linguistics, 38(4):799–825.

[50] Chi-kiu Lo. 2019. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), pages 507–513.