Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

WORD LEVEL ENGLISH TO HINDI NEURAL MACHINE TRANSLATION

[1a]NOMI BARUAH, [2a]AURANGZEB KHAN, [3a]DHANOJIT RAY,[4a]SAHID AFRIDI,[5a]RANDEEP BORAH, [6a]RAKHEE D. PHUKAN, [7a]ARJUN GOGOI

[a]Dibrugarh University, Dibrugarh-786004, India

E-mail: [1a]baruahnomi@gmail.com, [2a]dkhan0159@gmail.com,
[3a]dhanojitray2014@gmail.com,[4a]sahid.afridi6@gmail.com,[5a]randeep89borah@gmail.com,
[6a]rakheephukan54@gmail.com, [7a]gogoiarjun27@gmail.com

ABSTRACT

In today's world English is considered as important language across the Globe. Many resources are available in English language on the internet, which is not easily understandable , so its necessary that English language need to translate into the local languages of India so that the people of India can easily understand the enormous amount of English resources. As the information is of large amount so its not possible to keep translating things from one language to another manually. Thus its very important to translate the given text or information from one language to another automatically and effectively. This paper discusses about Neural Machine Translation(NMT) for converting English text to Hindi text. Neural machine translation(NMT) is one of the most recent and effective translation technique amongst all existing machine translation systems. In our experiment we have tested using 4 different model on OPUS, IIT-Bombay English-Hindi parallel corpora contains nearly 1084157 sentences and we have been able to get quite good results in terms of BLEU score while comparing to other available English to Hindi Neural Machine Translation model.It has achieved satisfactory score of 21.07,22.08,23.45 and 23.44 (in terms of percentage) for 2-layer, 4 Layer, 2 Layer (Bidir) and 4 layers (Bidir) LSTM respectively. Also,the accuracy of the system is compared with 4 existing machine translation system available in the internet for English to Hindi.Human evaluation of the systems is done based on five parameters and our system outperforms all the others.

KEYWORDS: *BLEU Score,Byte Pair Encoding, English-Hindi, Machine Translation, NLP, NMT.*

1. INTRODUCTION

The task of automated translation from one language to another has undergone rapid advancement due to the emergence of deep neural networks. Neural networks have been studied for machine translation in the 20th-century[1]. However, very recently it has reached state-of-the-art performance [2] with large scale deployment. In the Machine Translation (MT) community, a neural network based model for machine translation is referred to as Neural Machine Translation (NMT), where a sequence-to-sequence (seq2seq) [3] model is most Although Statistical Machine Translation (SMT) has been successful in the community in the last decade, however, the complete pipeline gets complex with the addition of more features, saturating the translation quality. This limitation of SMT and the success of deep learning has led to a focus on NMT approaches for machine translation in the MT community. Typically, the NMT consists of an encoder and a decoder. The first network, the encoder, processes a source sentence (e.g., English) into a vector (i.e., also referred to as a context vector or thought vector). A second network, called the decoder, uses this vector to predict the words in the target language (e.g., Hindi). Traditionally, NMT uses a different variant of Recurrent Neural Networks (RNNs), however, other architectures such as a Convolutional Neural Network (CNN) can also be used for the encoder. The advantage of NMT is that it learns mapping from the input to the output in an end-toend fashion, trained in a single big neural network. The jointly learns the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameters in order to maximize the performance of the translation output [5]–[7], which also requires minimum domain knowledge. In addition, similar to Statistical Machine Translation (SMT), NMT does not need to tune and store different models such as the translation language, and reordering models. The study of [8] reports that the NMT models require only a fraction of the memory needed by traditional SMT models. Since NMT emerged, it has been providing state-of-the art performance for various language pairs, however, the literature also reports its limitations, such as dealing with long sentences [8]. In order to deal with said issues Attention based mechanisms have been introduced, in which the model jointly learns to align and translate.. It is based on self-attention. The literature with NMT techniques report higher performances for resource-rich languages such as English to German [9] and English to French [10]. Compared to resource rich languages the literature of NMT for the English-Hindi language pair is relatively sparse. More details of the current state-of-the-art can be found in the next section. In this study, we aim to shed light on this area. Our contributions include, (i) conducting experiments using different NMT approaches, (ii) consolidating publicly available data from different sources and evaluating them using these approaches.

In recent times internet has developed vastly due to which different types of resouces are being available in English language on the internet, which is not easily understandable. Its quiet getting necessary that English language need to translate into the local languages of India so that one can easily observe and understand it.

In this paper, we experimented Neural Machine Translation (NMT) for converting English text to Hindi text as its very important to translate the given text or information from one language to another automatically and effectively in recent times.

The structure of this paper is as follows : Section 2 discuss the related works of NMT, section 3 describes the methodology of English to Hindi NMT, section 4 describes the working of the model, section 5 discuss the experiments and results and finally it concludes with the section number 9.

2. RELATED WORK

As of now there is very limited amount of works done in Neural Machine Translation from English to Hindi Language and to get a detailed idea about the proposed model we have considered various neural machine translation techniques and a few are discussed below and shown in table 1.

2.1 Neural Machine Translation in Non-Indian Languages

Yonghui Wu et al. [1] showed the power of GNMT. It contains eightlayer encoder-decoder architecture. GNMT changed the machine translation game with it is complicated architecture that requires huge GPU power for training the neural network. GNMT also known as zero shot translation. It can translate many intermediate language without training directly. Suppose we trained the model on English to French language pair and French to Hindi language pair, then it can translate English to Hindi, that is the main power of GNMT. To improve BLEU score even more , they have used reinforcement learning, though they achieved more bleu score but it has not improve human evaluation score. Google translation system was best at the time of deployment with highest BLEU score on WMT-14 English to French dataset. [2] Biao Zhang et al. have experimented on a variational model that learns conditional probability distribution of words for neural machine translationan end to end trainable model with encoder decoder architecture. This model is different from vanilla architecture. It generates target translation based on hidden state of encoder alone. This model uses latent variable to lead the creation of target translation in place of decoder hidden state and previous state input. To carry out an efficient posterior inference and large-scale training, they created a neural posterior approximator constrained on both the target and the source sides, and equip it with a reparameterization, method to predict the variational lower bound. [3] Yukio Matsumura et al. have experimented on encoder-decoder-reconstructor based model for neural machine translation using back translation. In their experiment, they chosen the best forward machine translation model in the same way as Bahdanau et al. [4], and then fine tuned bi directional translation model, which improved experiment result. Their above model showed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

that it achieved significant improvement in BLEU scores in Chinese-English translation work. They have also verified that their re-implementation also shows the same proclivities and reduces the difficulties of repeating and missing words in the translation on English-Japanese work too [7]. The main problem with neural machine translation is open vocabulary problem. Minh-Thang Luong et al. [5] have created a new word character level architecture to achieve open vocabulary neural machine translation solution. They kind of combined basic neural machine translation with character level neural machine translation to solve rare or unknown word problem. When the model found an unknown word it takes consideration of character of that word. The main advantage of this model is it is faster and efficient to train and also never produces unknown word. From analysis it is found that the model can learn and generate well-formed words.[5] Encoder decoder based model with attention is certainly not a new architecture. But Mingxuan Wang et al. [6] experimented on a model based on sequence to sequence architecture with attention, self-attention and some form of recursion which certainly produced an improved result. To make the model more efficient data augmentation method was applied with re ranking. Ensemble and re ranking techniques helped the model to achieve the best score. They have used the following methods- (a) back translation on target dataset (b) joint training of the T2S and S2T systems (c) knowledge filtration with teacher networks, R2L teacher networks, and ensemble teacher networks (d) system combination and re ranking. [6] Simple baseline neural machine translation can produce better result than statistical machine translation. Zhongyuan Zhu [7] has experimented neural machine translation model with data reordering. It is a pre-processing step to make neural machine translation more efficient. They experimented their model with the assumption that similar sentence structure will lead to better translation. This method certainly improved SMT architecture. But it is proven through this experiment that pre reordering of sentences hurt the model slightly. They provide a qualitative analysis demonstrating error pattern in neural machine translation[7]. Recurrent Neural Network is old version of neural network. There are some up gradation of neural networks for different purposes. We have seen many better version of RNN like LSTM, GRU in neural machine translation task. Kyunghyun Cho et al. [8] have compared the translation efficiency of RNN encoder decoder based architecture with a newly created gated recursive convolutional neural network architecture. RNN based model gives good performance on short sentences but the performance degrades quickly with sentence length and more number of unknown words found in a sentence. [8] Comparison helps us understand why some architecture works better than others. SMT was dominant at the time it was proposed. But when neural machine translation came it was more effective in translation than SMT. Even simple neural machine translation model with low data resource performs better. Praveen Acharya et al. [11] compared SMT and NMT[11]. Laskar S.R. et al. [24] have experimented on encoder-decoder based model for neural machine translation where the the exact challenge is to provide a precise MT output although they have been able to get with a Bleu Score of 20.37. Saini S. et al. [25] have experimented with eight different architecture combinations of NMT for English to Hindi and compared their results with conventional machine translation techniques with a Bleu Score of 18.12. Stahlberg [40] proposes a work where he trace back the origins of modern NMT architectures to word and sentence embeddings and earlier examples of the encoder-decoder network family and concluded with a survey of recent trends in the field.

2.2 Neural Machine Translation in Indian Languages

Himanshu Choudhary et al. [12] have given a new neural machine translation architecture. They have used word vectorization with byte pair encoding(a data compression technique)on English-Tamil language pair. It is a better translation system that overcomes out of vocabulary problems. Although it is not practical to use in real life translation work but it has showed some improved result over google translation system. They have used most popular translation evaluation method BLEU score to evaluate their system. This paper is also important in the perspective to the point, why pre-processing is important. [9] Sandeep Saini et al. [13] have given a different architecture setting for their machine translation task in Hindi language. They have tuned their setting with eight various architecture combinations of neural machine *translation for English-Hindi language pair and compared their outcomes with existing machine translation systems. This system showed how to make an efficient translator with less amount of data. It is particularly important for low resource language pairs. From their experiments they concluded their satisfactory results.[14] Sukanta Sen et al. provided a restricted translation model because the model heavily relies on dataset and language pairs. They have used transformer architecture as a base model with subword neural machine translation(subword NMT). They have also showed the effect of back translated data on a translation system. As a result, they have concluded that adding back translated data on original language pairs can make an efficient system.[14] In this study Md. Arid Hasan et al. [15] have explored different neural machine translation architecture. Their system uses bidirectional long short term memory (LSTM) and transformer based neural machine translation, to translate the Bangla to English language pairs. For their experiments, they have used different datasets. Their translation system outperforms the existing translation system on the Bangla-English translation pair. Their work showed the importance of good quality dataset, that can make an improved translation system. Promila Bahadur et al. [16] have showed the complete frame of work for rule based machine translation system. They experimented on Sanskrit-English language pair, as target and source language. Their system supports both Sanskrit and English grammar such as noun, adjective, verb etc. To verify robustness of the system(basically rules' robustness), E Trans translation system took hundreds of sentence samples of various types, as the compound and simple sentences of interrogative, armative and imperative types in passive and active voice. they have taken sentences from all the three tenses i.e. future, present, past . It was their assumption that this type of technique can be used for translation of similar language pairs. The rule based translation system can be used to translate various types of documents in English to Sanskrit [16]. Revanuru et al. Also proposed NMT where they compare the performances of NMT models with system using automatic evaluation metrics such as UNK Count, METEOR, F-Measure, and BLEU and found out that NMT techniques are very effective for machine translations of Indian language pairs [35]. Choudhary et al. proposed a novel NMT model using Multihead self-attention along with pre-trained Byte-Pair-Encoded (BPE) and MultiBPE embeddings to develop an efficient translation system that overcomes the OOV (Out Of Vocabulary) problem for low resourced morphological rich Indian languages which do not have much translation available online. They used the BLEU score for evaluating system performance and found out that proposed translator (24.34 and 9.78 BLEU score) outperforms Google translator (9.40 and 5.94 BLEU score) respectively [36]. Singh et al. have proposed and presented the machine translation system for translating Sanskrit to the Hindi language. The developed technique uses linguistic features from rule-based feed to train neural machine translation system and the results show that proposed and developed approach outperforms earlier work for this language pair [37]. Pathak and Pakray have proposed NMT where they have trained, tested, and analyzed NMT systems for English to Tamil, English to Hindi, and English to Punjabi translations. Predicted translations have been evaluated using Bilingual Evaluation Understudy and by human evaluators to assess the quality of translation in terms of its adequacy, fluency, and correspondence with human-predicted translation [38]. Shah et al. have proposed and compared the performance of NMT model with automatic evaluation matrices such as BLEU, perplexity and TER matrix. The comparison of network with Google translate is also presented where it outperformed with a margin of 6 BLEU score on English-Gujarati translation [39].*

Table 1: Comparison of different NMT Models

Title	Corpus	Architecture	Domain	Bleu Score
Google's Neural Machine Translation the Gap[1]	WMT En-Fr dataset (36 M sentence)	Encoder + attention bridge + Decoder (zero shot translational)	NMT	38.95

Variational Neural Machine Transla-tion[2]	Chinese English dataset (2.9 M sen-tences)	Variational Neural Machine Translation + unk replace	NMT	19.58
English - Japanese with Encoder Decoder Reconstructor[3]	ASPEC(827,188) sen-tence	Baseline NMT + Reconstruc-tor(jointly training)	NMT	26.04
English Japanese Neural Machine Translation with Encoder Decoder Reconstructor[3]	NTCIR(1,169,201)	Baseline NMT + Reconstruc-tor(jointly training)	NMT	29.04
Sequence to Sequence learning with Neural Networks[4]	WMT'14 English to French dataset (12 M sentences)	Baseline NMT + Rescoring the baseline 1000 best with an essemble of 5 reversed LSTM's	NMT	36.5
Evaluating Neural Machine Translation on English Japanese task[7]	English Japanese dataset (1.5 M sen-tences)	Ensemble of 2 LSTM search+ UNK replacing + Sys-tem combinations+ 3 pre-reordered Ensembles	NMT	35.97
Learning Phrase Representations us-ing RNN Encoder Decoder statistical Machine Translation [8]	The Bilingual Corpora include Europarl (61 M words)	Baseline NMT + word penalty	NMT	34.54
A comparative study of SMT and NMT[11]	Nepali National Cor-pus(6535 sentences)	Baseline Neural Machine Translation	NMT	3.28
Neural Machine Translation for English-Tamil[12]	EnTum V2.0 and 0pus(1,83,451) sen-tence	Byte Pair Encoding + En-coder + attention + Decoder	NMT	8.33
Gujrati English news Translation task[14]	English Gujarati sen-tence pair 155,798 sen-tences	Transformer based NMT	NMT	4.0
Neural Machine Translation for the Bangla-English Lan-guage Pair[15]	Bangla corpus	BiLSTM based NMT archi-tecture	NMT	18.73
A	Sanskrit	BiLSTM	NMT	18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Complete Frame-work for English To Sanskrit Machine Translation[16]	corpus	based NMT archi-tecture		
Neural Machine Translation of rare words with sub-words[17]	English German dataset (4.2 M sen-tences)	Byte Pair Encoding + Se-quence to Sequence architec-ture+ attention	NMT	22.8
English to Hindi Multi-modal Neural Machine Translation and Hindi Image Captioning[24]	English to Hindi hav-ing 28,929 instances	Attention-based decoder	NMT	20.37
Neural Machine Translation for En-glish to Hindi[25]	English-Hindi parallel corpus from Institute for Language	Attention-based decoder encoder-	NMT	18.2

3. METHODOLOGY

3.1 Softmax

The softmax function is a type of function (squashing function) that squashes the input to one of the ends of a small interval. The output of the function is limited into the range 0 to 1 by the squashing function. Further the output is directly expounded as a probability. Additionally, softmax capacities are multi-class sigmoids, which means they are utilized in deciding probability of various classes at a time. As we can say that the output of a softmax function is expounded as a probability (i.e. their sum must be equal to 1), the softmax layer is generally the last layer which is applied in neural network functions. It is necessary to keep a note that the number of nodes in softmax layer must be same as the output layer.

3.2 RNN

RNN (Recurrent Neural Network)[30] is a kind of neural network in which the output generated from the previous stage is taken and fed as input into the current stage(step). In the earlier neural networks, the inputs and outputs were not dependent on each other, but in situations where there is requirement of predicting the next word of a given sentence, the earlier words are required and hence it is very important to recall the previous words. Hence RNN appeared, which comprehended this issue with the assistance of a Hidden Layer. The primary and most significant element of RNN is Hidden state, which recalls some information about a sequence. RNN have a ” memory” which recollects all data about what has been determined. It utilizes the similar parameters for each input as it plays out similar task on all the data sources or hidden layers to deliver the output. This diminishes the unpredictability of parameters, in contrast to other neural systems.

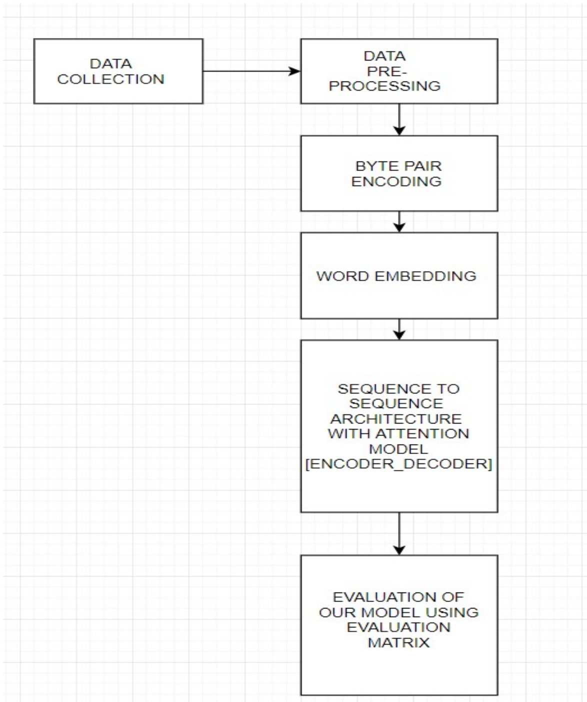

Figure 1: Architecture Of Our Proposed Model.

3.3 Data Collection

3.3.1 Steps

The description of the steps are given in Figure 1.

1. Collection of English-Hindi Translated sentence pairs from different sources.

2. Data Pre-processing:

Reiteration of source sentences, translated sentences are taken out and tokenization is finished.

3. We utilize this BPE calculation for word division (word Segmentation).

4. Sequence to sequence mechanism (architecture) is utilized to discover relationship between two distinctive language sets .

5. Our outcome is contrasted and human interpreted

Assortment of English-Hindi Translated sentence sets from various sources.

3.3.2 Pre-processing of text

(a)Removal of :

1. Repetition of sentence with same source and same targets.
2 .Repetition of sentence with same source and various targets.
3 .Repetition of sentence with various source and same targets.

(b) Tokenization of words.

3.4 Byte Pair Encoding

Byte Pair Encoding, BPE is a straightforward data compression method. It replaces most successive pair bytes in an arrangement with single unused byte. We utilize this algorithm for word division. By consolidating regular sets of bytes we blend contracts or character successions. NMT symbols interpretative as sub-words units and systems can translate and make the new word based on sub-words. BPE will assist with parting compound word and suffix, prefix partition which is utilized for making new expressions of Hindi language. We will utilize BPE alongside word embedding [17] .

3.5 Word Embedding

Word implanting (Word Embedding) is the aggregate name for a lot of language displaying and include learning strategies in natural language processing (NLP) where words or expressions from the vocabulary are mapped to vectors of genuine numbers. Reasonably it includes a numerical implanting from a space with numerous measurements per word to a constant vector space with a much lower measurement. Word implanting is a method of speaking to words on a vector space where the words having same importance have comparative vector portrayals. Each word from vocabulary is on represented to in many measurements. Typically prepared word implanting are utilized and with the assistance of transfer learning words from jargon are changed over to vector [8].In our model, we utilized Fast Text word vectors to convert over English and Hindi vocabulary into vectors.

3.6 LSTM

In a feed-forward neural system, inputs are fed into the system, and the system gives an output. In administered learning, the yield (output) would be a class id or a label. While a Recurrent Neural Network (RNN) not just takethe current info that is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

taken care of into, yet in addition the data sources utilized already to create the yield. RNNs are valuable learning successive information. It has a weight framework that associates shrouded state to concealed state at previous time step. The successive data is protected in the shrouded state. To group consecutive info, RNN rely upon the back propagation of error and slope(Gradient) Descent. The serious issue looked by RNNs is that the vanishing gradient issue. The gradient exponentially disappears as its backpropagation through time. By utilizing LSTM organize, we can overcome this issue to some extent. LSTM network is a unique sort of RNN with LSTM blocks or units. These LSTM units save the errors that can be back propagation through time. They permit RNN to learn over many time ventures by keeping up a more consistent error. There is something many refer to as gated cells in a LSTM unit, which controls the progression of information in the cell. In a gated cell, data can be put away, peruse and compose tasks can be performed. There is input gate , yield(output) gate and forget door in a unit. These (gates)entryways have their weights. Rather than utilizing a steady long term memory, these gated cells utilize the component of overlooking pointless data and putting away data which is helpful. The choice to pick what data is to discard is made by forget gate, which is a sigmoid layer. It takes ht 1 and xt , as input and outputs a number between 0 and 1 for each number in the cell state Ctt 1. Forget gate ft is defined as:

$$it = \sigma g(wi.[xt\ ,\ ht\ 1] + bi) \text{...........(ii)}$$

$$Ct = tanh(Wc.[xt\ ,\ ht\ 1] + bc \text{...........(iii)}$$

Now the old cell state $Ct\ 1$ is updated into new cell state Ct

$$Ct = ftCt\ 1 + itCt \text{ ....(iv)}$$

Finally, the output will be based on the cell state. First a sigmoid layer is run ,which decide which part of the cell state are going to output then, the cell state is put through and multiplied it by output of the sigmoid gate.

$$ot = \sigma g(Wo.[xt\ ,\ ht\ 1] + bo) \text{........(v)}$$

$$ht = ottanh(Ct) \text{.........(vi)}$$

4. WORKING OF THE MODEL

The working of the model is described in figure 2.Here we can see that the Sequence to Sequence architecture with the attention mechanism has been explained we have taken an English sentence " You are good/\texthindi{□□ □□□□□ □□□"} which is to be translated in Hindi. In this architecture the Encoder part is very same as traditional Sequence to Sequence architecture. In " You are Good/\texthindi{□□ □□□□□ □□□} we have each word denoted as h1, h2, h3. We are going to use it as each state value h1, h2, h3 and the final output is h3 here. Since the final output is h3, the next fully-connected layer will take this h3 as input. Also we will use every encoder's RNN cell state value so we can see the fully connected layer has h1, h2, h3 and also we have the h3 again because this was from the previous stages as state value. From here we get three scores s1, s2, s3 because we had three state value here. We do softmax on it that means the output of the softmax will have the probability value and the numbers we get 0.1 for " you/\□□" 0.2 for " are/\□□□" and 0.3 for " good/\□□□□□" we call it attention weights, that means we want to attention or put focus on " You/□□" 10 percentage " are/□□□" 20 percentage and " good/□□□□□" 30 percentage and then we get the contextralized vector (h1*0.1+h2*0.2+h3*0.3).

We are highlighting " You/□□" in this context vector and this contextralized vector is coming to the decoder's input and also this is decoder's starting times so we give the start signal here, this RNN cells are putting ?? and also are putting counter state value here, this counter state value shows decoder hidden state 1(dh1).Lets go to the next step, in this step decoder's hidden state is coming to the fully connected network also we are again using h1, h2, h3 we see the difference here. Lastly we use h3 for the second fully connected layer in the fully connected layer but this time we use dh1 in the fully connected layer because this is the previous state from the decoder but we keep h1, h2, h3 because we still want to use this hidden state from the encoder and we get the softmax value here.

Again this attention weight 0.1 for You/□□ 0.2 for are/\□□□ 0.3 for good/\]□□□□□ and this second contextralized vector will be (h1*0.1+h2*0.2+h3*0.3).Eventually it means we want to highlight Good/]□□□□□ and You 10

Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

percentage and 20 percentage for are/□□□ here this comes to the decoder second RNN cell and also the input will be the previous output of the decoder which is You/□□ after doing that the second RNN cell will output good/□□□□□ here also output the state dh2 and we can see the dh2 is coming to the fully connected network where we again use h1, h2, h3 because we want to have attention weight here.

After the softmax the attention weight for " are/□□□" is 20 percentage and " You/□□" has 10 percentage and " good/□□□□□" has 30 percentage only so context vector 3 will focus on are this cv3 is coming to the decoder RNN cell with the output from the previous decoder's RNN cell and eventually it will output are/□□□ here and we will see the end.

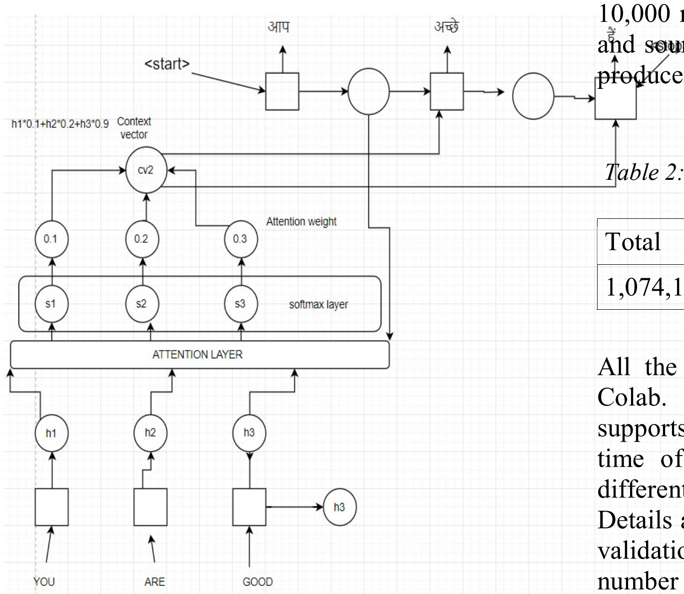

Figure 2: Working Of The Proposed Model.

5. RESULTS

5.1 Dataset

For our experiment we needed large datasets of translated sentence. Table 5 describes the number of sentences pairs that are collected English to Hindi translated sentence pair from OPUS[18]. It is an increasing collection of translated texts from the web, also we used IIT-Bombay English-Hindi parallel corpora provided by the organizer[19]. All the datasets contain sentences from various domains like TED talks, Hindi-English wordnet linkage, Tourism, Travel , Different Indian Government websites , Book Translations (Gyaan-Nidhi Corpus). Total around 1084157 sentences are collected for our Neural Machine Translation System experiment. For Neural Machine Translation or deep learning model or any other machine learning model, data is very essential. The result of good translation heavily depends upon good quality of Hindi- English translated sentences. To make sure of good quality dataset, we applied mainly two approach to clean the collected datasets - i) Manual cleaning. ii) Automatic cleaning. Figure 3 describes the data cleaning of our work.

5.2 Training Details

When the dataset is preprocessed, It is divided into training- 1,021,215 testing-42,942 and validation-10,000 respectively after shuffling. Then the target and source files are given into the encoder layer to produce the word vectors of the sentences.

Table 2: No. Of Sentences Used For Training ,Validation And Testing.

Total	Training	Validation	Test
1,074,157	1,021,215	10,000	42,942

All the experiments were carried out on Google Colab. Google Colab is a cloud service and it supports GPU [21]. Because we used GPU ,training time of the neural network for our dataset for different architectures was in only few hours. Details about number of sentences used in training, validation and testing is shown in Table 2. Total number of steps taken to train the dataset was 1,00,000. We perform all experiments based on Simple and Effective Hindi-English Neural Machine Translation Systems paper [22]
For all of our experiments, we used OpenNMTpy toolkit [20] .Our network layer contains LSTM, a modified version of RNN unit. We used Bi-directional LSTM encoder and a unidirectional LSTM for decoder along with global attention mechanism. We kept 4 layers in both the encoder and decoder with embedding size set to 512. The batch size was set to 128 and a dropout rate of 0.3. For optimization, we used Adam optimizer for all our experiments. The result of the execution of the experiment is mentioned in table 3 and the parameters used in the training are mentioned in table 4.

Table 3: Bi-LSTM Bi Directional LSTM, BPE Byte Pairencoding Optimizer ADAM Attention Type Mlp

Total sentences	2 layers or Bi-LSTM or Attention or BPE (60,000) or ADAM	4 layers or Bi-LSTM or Attention or BPE (32,000) or ADAM	4 layers or Bi-LSTM or Attention or BPE (60,000) or ADAM
1074157	14,390 secs	16340 secs	18570 secs

Table 4: Parameters Used In The Training.

Train steps	100000
Src word vec size	512
Src word vec size	512
Src word vec size	Brnn(BI-DIRECTIONAL LSTM)
Enc layers(total en- 4 coder layers)	4
Dec layers (total de- 4 coder layers)	4
Rnn size (size of RNN)	500
Rnn type(type of LSTM RNN)	LSTM
Global attention	M1p
optim (optimizer)	AdaM
learning rate	0.001
Batch size	128
Dropout	0.3
learning rate decay	0.5

Table 5 : Number Of Sentence Pairs Collected For Pre-Processing

Dataset	Total Sentences
IIT-Bombay [31]	788098
OPUS [32]	296059

5.3 Experiment

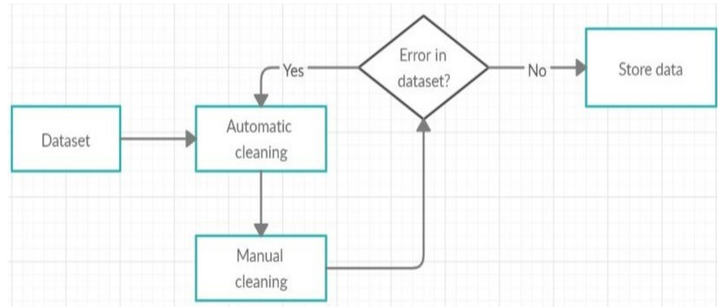

Figure 3: Flow Diagram Of Data Cleaning Method.

As figure 3 suggest, from our dataset we cleaned it with automatic method then observed error pattern manually if we find any error, we repeated above step otherwise we stored cleaned sentence into the dataset.

In our corpus, there existed many repeated sentences, which outcomes the wrong results (may be high or low) after dividing into train, test, and validation sets, as some of the sentences occur both in train and test datasets. Thus, it is important to clean, analyse, and correct it before using for experiment. We discovered the following main problems in the corpus.

1.Repetition of sentences with same source and same target
2. Sentences with same source and different translation.
3. Sentences with wrong translation pair.

To solve the first problem we took all the distinctive pairs from the dataset and eliminated the repeating ones. For the second and third case, we completely removed those sentences which are repeated more than once because in the second case our model may get confuse and results the wrong output. For third case it is obvious to remove the wrong translated sentence pair. Finally after working on all these small but effective preprocessing such as removing sentences with the length smaller than 1, removing non translated words in the target sentences, removing erroneous translations and extra punctuations, we got our final dataset of 1074157 parallel sentences which was cleared from 1084157 sentences.

5.4 Evaluation Metric

The bilingual evaluation under study(BLEU score) is a technique to calculate the difference between human translations and machine[23]. The method works by calculating and matching n-grams in result translation to n-grams in the reference text, where unigram would be each space separated symbol(or token) and a bigram would be comparison of each token pair and so on. The comparison is done regardless of token sequence. This technique is moderation of a simple precision method.

5.5 Comparison of the proposed model

We have assessed our system using BLEU score. In each configuration, BLEU scores are found to be different and table 6 shows the BLEU score for different configuration.

Table 6 : BLEU Score Of Different Configuration

Model	BLEU(in percentage)
2 layers+Bi LSTM+Attention+BPE(60000)+ADAM	21.07
4 layers+Bi-LSTM+Attention+BPE(32000)+ADAM	22.08
4 layers+Bi-LSTM+Attention+BPE(60000)+ADAM	23.4
4 layers+Bi-LSTM+Attention+BPE(90000)+ADAM	23.44

We mainly analyzed our model based on translation quality.We applied two approach to analyze our model.

1. BLEU score, we have compared our proposed model with some of the available machine translation system available in the internet.For that we have selected 150 sentences randomly and divided into 5 data sets putting 30 sentences in each. Then using different translators available on the web, we translated each data set.The system which we have compared are Anusaaraka model [26],CDAC model [27],Google Translate [28].The BLEU score result is shown in the figure 4 in the bar graph.Also we have compared our proposed model with two previously published papers who had done neural machine translation from English to Hindi. They are [24] and [25] .It is seen that in table 8 our proposed model performs better than previous work done.

2. Comparing translated sentences by giving some rating on the basis of how correctly it translate a sentence. Every sentences were categorized either to be completely understandable and grammatically correct, completely understandable, mostly understandable, or not understandable. Sentences that are completely understandable and grammatically correct must carry the same idea that is conveyed in the original sentence and must be correct grammatically. Sentences in the completely understandable and mostly understandable types may not be correct grammatically, but should not be too deviated from original meaning. Sentences that are in the not understandable section have no influence in this method. We assign a score from 0 to 1 for each category: 1.0 for completely understandable and grammatically correct, 0.75 for completely understandable, 0.5 for mostly understandable, and 0.0 for not understandable. We wanted to compare our model with other machine translation system available on the internet. We took more than 50 sentences from our test dataset and applied above mentioned method on translated sentences to analyze. The results are shown in table 7. We passed our decoder to 55 English sentences. After scoring each translation using the method described earlier, we obtained the results shown in

Table 7. Out of 55 translated sentences from English to Hindi, 29 were completely understandable and grammatically correct, 14 were completely understandable, 8 were mostly understandable, and 4 were not understandable. The average score of our model of is 79.09 percentage.

Table 7: Human Evaluation Score Comparision Among Our Model,Google Translate,CDAC Model And Anusaaraka Model.

	CDAC in percentage	Anusaaraka in percentage	Google in percentage	Our Model
Fullyder stand-able,correct grammar	8	18	26	29

Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Fully understandable	18	19	11	14
Mainly under-standable	21	14	15	8
Not understandable	8	4	3	4

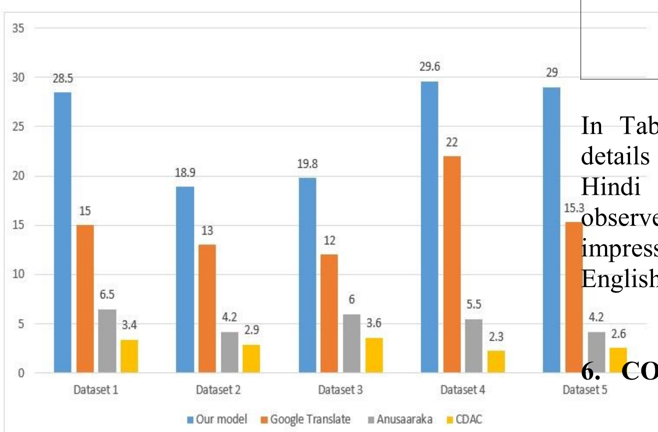

Figure 4: BLEU Score Comparision Among Our Model,Googletranslate ,CDAC Model And Anusaaraka Model.

Table 8 : Comparison Of Our Proposed Model To Other English To Hindi Neural Machine Translation In Terms Of BLEU Score

Model	2 Layer LSTM	4 Layer LSTM	2 Layer (Bi-dir) LSTM
Proposed model	21.07	22.08	23.45
Neural Machine Translation for English to Hindi [24]	6.86	17.12	18.1
English to Hindi Multi-modal Neural Machine Translation and Hindi Image Captioning [25]	20.37	12.57	11.77
Model	2 Layer LSTM	4 Layer LSTM	2 Layer (Bi-dir) LSTM

In Table 8 we have presented the Comparison details of our proposed model to other English to Hindi Neural Machine Translation and it is observed that our model has performed impressively well than from the prior works of English to Hindi Neural Machine Translation.

6. CONCLUSION AND FUTURE SCOPE

In this paper, we experimented with neural machine translation on English Hindi sentence pair and showed that neural machine translation with Byte Pair Encoding and word embedding is better translation method than complex translation methods on Indian languages.

Besides that, we have also showed the comparison between the prior works and our proposed model and found out that our proposed model performs better than all the prior works available in English to Hindi Neural Machine Translation since we achieved fairly good results.

In future would like to train of large and rare sentences using smaller data sets. We would also like to explore NMT for other Indian language pairs as well. Since the grammar structure for many of the Hindi languages is similar to each other, we expect the higher order of BLEU scores in future.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] Y. Wu., M. Schuster , Z. Chen, V.l Quoc , M. Norouzi, "Google's Neural Machine Translation System Bridging the Gap between Human and Machine Translation",(2016)

[2] B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang,"Variational Neural Machine Translation",*Proceedings of the Conference on Empirical Methods in Natural Language Processing*,P.521–530,(2016)

[3] Y. Matsumura, T. Sato, M. Komachi, "TMU Japanese–English Neural Machine Translation System using Generative Adversarial Network for WAT 2018",*32nd Pacific Asia Conference on Language, Information and Computation The 5th Workshop on Asian Translation*,P.1112-1119,(2017)

[4] D. Bahdanau, K. Cho, Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate",*Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, P. 1–15,(2015)

[5] M.T Luong, and C.D Manning,"Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models",*Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, P. 1054–1063,(2016)

[6] M. Wang , L. Gong , W. Zhu, J. Xie , C. Bian , "Tencent Neural Machine Translation Systems for WMT18",(2018)

[7] Z. Zhu ,"Evaluating Neural Machine Translation in English Japanese TasK",*Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*,P.61–68,(2015)

[8] K. Cho,B.V Merrienboer ,D. Bahdanau ,Y. Bengio ,"On the Properties of Neural Machine Translation Encoder-Decoder Approaches",*Proceedings of SSST @EMNLP Eighth Workshop on Syntax Semantics and Structure in Statistical Translation* ,(2014)

[9] M. Popovic ,"Comparing Language Related Issues for NMT and PBMT between German and English",*Prague Bulletin of Mathematical Linguistics*,P.209–220,(2015)

[10] V.S.R Middi , M.R Raju, T.A Harris ,"Machine translation using natural language processing",*MATEC Web of Conferences*,https://doi.org/201927702004,(2019)

[11] P. Acharya, B.K Bal ,"A Comparative Study of SMT and NMT Case Study of English-Nepali Language Pair",*The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*,http.//dx.dor.org/10.21437/SLTU.2018-19,(2018)

[12] H. Choudhary , A.K Pathak , R.R Shah , P. Kumaraguru ,"Neural Machine Translation for English-Tamil",*Proceedings of the Third Conference on Machine Translation Shared Task Papers*,P.770–775,(2018)

[15] S. Saini , V. Sahula ,"Neural Machine Translation for English to Hindi",*2018 Fourth International Conference on Information Retrieval and Knowledge Management*, P. 1-6,http.//dx.dor.org/110.1109/,(2018)

[16] S. Sen , K.K Gupta , A. Ekbal , P. Bhattacharyya , "IITP-MT System for Gujarati-English News Translation Task at WMT" ,*Proceedings of the Fourth Conference on Machine Translation* ,2:407–411,(2019)

[17] M.A Hasan , F. Alam , S.A Chowdhury , N. Khan ,"Neural Machine Translation for the Bangla-English Language Pair", *22nd International Conference on Computer and Information Technology (ICCIT),*http.//dx.dor.org/10.1109/,(2019)

[18] P. Bahadur , A.K Jain , D.S Chauhan , S. Etran , "A Complete Framework for English To Sanskrit Machine Translation",*The Science and Information (SAI) Organization*, P.52-5,(2012)

[19] R. Sennrich , B. Haddow ,"Neural Machine Translation of Rare Words with Subword Units",P.1715-1725,(2016)

[20]corpus. http://opus.nlpl.eu/ accessed on 12th march,2020

[21] A. Kunchukuttan , P. Mehta ,P. Bhattacharyya ,"The IIT Bombay English-Hindi Parallel Corpus Language Resources and Evaluation Conference",*Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*,(2018)

[22] OpenNMT-py is a community developed project.https://opennmt.net/OpenNMTpy/ accessed on 12th march,2020

[23] Colaboratory, or 'Colab' for short, allows you to write and execute Python in your browser https://colab.research.google.com/notebooks accessed on 12th march,2020

[24] D.M Sharma , V. Goyal ,"LTRC-MT Simple and Effective Hindi-English Neural Machine Translation Systems at WAT",P.137-140,(2019)

[25] A. Benabbou ,M. Brour ,"Arabic text language into Arabic sign language neural machine translation",*Journal of King Saud University*

Computer and Information Sciences,P.1-11,(2019)
[26] S.R Laskar , R.P Singh , P. Pakray , S. Bandyopadhyay ,"English to Hindi Multi-modal Neural Machine Translation and Hindi Image Captioning",*Proceedings of the 6th Workshop on Asian Translation* ,P.62–67,(2019)
[27] S. Saini , V. Sahula ,"Neural Machine Translation for English to Hindi",*Fourth International Conference on Information Retrieval and Knowledge Management*,DOI: 10.1109/INFRKM.2018.8464781, (2018)
[28] http://sampark.iiit.ac.in/anusaaraka/ accessed on 12th march,2020
[29] http://tdil-dc.in/mt/common.php accessed on 12th march,2020
[30] https://translate.google.co.in/ accessed on 12th march,2020
[31] K. Shim, M. Lee , I. Choi ,Y. Boo , W. Sung,"SVD-Softmax: Fast Softmax Approximation on LargeVocabulary Neural Networks",*31st Conference on Neural Information Processing Systems(*2017)
[32] S. Yang,Y. Wang , X. Chu," A Survey of Deep Learning Techniques for Neural Machine Translation",arXiv:2002.07526,(2020)
[33] A. Kunchukuttan , P. Mehta., P. Bhattacharyya ,The IIT Bombay English-Hindi Parallel Corpus
[34] The open parallel corpus,http://opus.nlpl.eu amp Time Stamp GMT and dd/mm/yyyy
[35] K. Revanuru, K. Turlapaty and S Rao, "Neural Machine Translation of Indian Languages", *10th Annual ACM India Conference.*
[36] H. Choudhary, S. Rao and R. Rohilla, "Neural Machine Translation for Low-Resourced Indian Languages", *LREC-2020*
[37] M. Singh, R. Kumar and I . Chana, "Improving neural machine translation for low-resource Indian languages using rule-based feature extraction", *Neural Computing and Applications .*
[38] A. Pathak and P. Pakray, "Neural Machine Translation for Indian Languages", *Journal of Intelligent Systems.*
[39] P. Shah and V. Bakrola, "Neural Machine Translation System of Indic Languages – An Attention based Approach", *International Conference on Advanced Computational and Communication Paradigms*
[40] F. Stahlberg, "Neural Machine Translation: A Review and Survey", *Journal of Artificial Intelligence Research (JAIR)*