ASM Sc. J., 13, Special Issue 2, 2020 for ICSCC2019, 31-45

Hybrid Machine Translation System Using Deep Learning

Muskaan Singh[1]*, Ravinder Kumar[2] and Inderveer Chana[3]

[1]CSED, Chandigarh University, Mohali, India

[2]Language Engineering and Machine Learning Research Labs, CSED,

Thapar Institute of Engineering and Technology, Patiala, India

[3]CSED, Thapar Institute of Engineering and Technology,Patiala, India.

Sharing of information and ideas require inter-language translation. It is one of the central application in most natural language processing applications. Sanskrit is an important language in the Indo-European family. A significant amount of research work in this language pair is required to open perspective in the computer science and computational linguistics domain. The main drawback in this domain is the technique used for developing i.e, rule-based which is not extendable to generic and huge domains. To overcome this problem, an efficient system is required to be developed which would cover various domains. Therefore, we have proposed a hybrid system combining the best of Neural Machine Translation (NMT) and Rule-Based Machine Translation (RBMT) is developed in this paper. It uses deep learning feature to overcome drawbacks of the existing systems. Experimental results show that the proposed neural model utilizes 99.99% accuracy. It is also evaluated that it has less response time and more speed than the existing rule-based systems.

Keywords: natural language processing; machine translation; neural machine translation; statistical MT; rule-based MT

I. INTRODUCTION

Natural Language Processing (NLP) is an open area of research which allows computers to understand the text in natural languages such as English, Hindi, Sanskrit, etc. The main aim of NLP is to develop models for computational purpose. Machine Translation System (MTS) is one of such applications of NLP (Bharati, 1995; Chatterji, 2009). It is a process of translating source language to target language using a computerized system. NLP helps to develop intelligent computer systems. MTS has various types and NMT is one from them. Neural Machine Translation (NMT) is an expressive approach for machine translation that provides remarkable progress as compared to rule-based and statistical machine translation (SMT) techniques, by eliminating its limitations. NMT uses both deep learning and representing learning. The major benefit to the method is that a particular system can be expert straight on the source and target text, no longer needful the pipeline of specific systems used in statistical machine learning. NMT is known as end-to-end systems also as it needs only one model for the translation.

India is a country of wide language diversity, with more than 1.3 billion people who have 22 official languages and more than 12 scripts; hence there is a need to provide the translation of the content from one language to another language. It is observed from various surveys that Sanskrit as a source or target language is in developing stages. Sanskrit is being the mother of almost all Indian languages. One of the main requirements in Sanskrit domain is to translate the life-transforming stories (epics), Vedas etc. to make them available in other languages, for the public at large. A major issue which arises in the implementation of the Sanskrit based machine translation system is the

*Corresponding author's e-mail: muskaan_singh@thapar.edu approach used in developing MTS for Sanskrit to Hindi translation system. Currently, the Sanskrit based system uses a dictionary and rule-based approach (Drummer and Aurelia, 2015). To remove this problem of extension to generic and huge domains, an efficient MTS is developed in our proposed work. Although NMT is a new approach to machine translation, it has produced promising results in the translation domain. It also has some limitations such as the amount of training data, word alignment and translation for long sentences etc. which we would overcome by forming it as a hybrid system along with rules. Thus, a hybrid system combining the best of NMT and RBMT has been developed in this paper.

The Sanskrit language translation is a challenging task in machine translation due to its linguistic richness. Increasing the amount of data causes an inconvenience in the translation of long sentences. Another issue addressed in this work is the alignment of words. It contains grammatical error along with the improper alignment of noun, verb etc. To overcome these drawbacks, we have been developed a hybrid model. This model contains various hidden layers, activations functions along with backpropagation. It makes the system more efficient and reliable by extracting all kind of useful information even from a huge dataset by processing the data many times. In the end, we get better accuracy as compared to the other existing models.

In section 2, the background of various machine translation models is discussed. In section 3, 4, 5 related work, proposed methodology and experimental details respectively have been discussed.

II. BACKGROUND

The section contains a brief explanation of the technologies and approaches used for developing the proposed system.

A. Rule-based Machine Translation (RBMT)

RBMT is one of the oldest approaches for Machine Translation (MT). It comprises of linguistic rules, dictionaries, lexicon, morphological analyser and generator, parser, POS tagger, chunker, transliteration module, tokenize, Word Sense Disambiguation and modules specific to language (such as in case of Sanskrit Sandhi-splitter required) (Forcada, 1997.)

RBMT follows three basic steps: Analysis, transfer and generation. As input text is provided to MT system, a sequence of steps followed are morphological analyser, POS tagger, Lexical selection, structure transfer, morphological generator and post-generator gives target or output text (Cho, 2014).

Human efforts and linguistic knowledge are required to develop each phase of RBMT, from rules to codes for different modules. It is easily extendable and maintainable. There are three approaches to RBMT: Direct Translation, Interlingua based Translation and Transfer based Translation (Cho K. B., 2014).

In the direct translation source text, words are matched corresponding to the dictionary and analysed structurally to morphological analysis which gives target language (Cho K. B., 2014). In Interlingua based approach Source language text is converted to an intermediary representation which is called Interlingua (not language-specific). From intermediary representation target language is generated. It is a suitable approach if the translation of multiple languages is to be performed. It is difficult to define Interlingua. The generator of the parser of the source text is independent of the target language generator (Dorr, 2004). This approach was more preferable or used by researchers. It is based on the structural divergence between source and target (Noone, 2003). It works in stages as follows (Nithya, 2013).

- Analysis: source language provided to the system is parsed for analysis.
- Transfer: source language parser representation is converted to target language.
- Generation: Target language morphological generator is used to generate the final target language. The section contains a brief explanation of the technologies and approaches used for developing the proposed system.

B. Neural-based Machine Translation System (NMT)

NMT approach is a new transpired approach in the field of MT (Machine Translation) (Sutskever, 2014) (Cho K.B., 2014). It is a new approach based on Neural Networks (NN) added to Statistical Machine Translation (SMT). In SMT parts of the sentence are trained (Word, Phrase etc.) whereas in the case of NMT it reads a sentence and gives output by training and building a large NN (Koehn, 2003). In SMT probabilistic model is calculated using S as source sentence and T as target sentence, maximum conditional probability is chosen as target sentence

$$\widehat{S} = arg \max_{s} \Pr(S|T) \quad (1)$$

In the case of NN, the translation model and language model are built from a corpus that provides translated target sentence corresponding to the given source sentence, which maximizes conditional probability. Direct learning for maximizing conditional probability has also been performed (Khan, 2011).

There are two different models of NMT: The RNN Encoder-Decoder (RNNenc) and gated recursive Convolutional neural network (grConv) (Cho K., 2014).

C. Hybrid Machine Translation

Combining the advantages of both RBMT and SMT, newer approach hybrid MT was introduced. It is based on rules and statistics or probability. Combination of both approaches can be performed in several ways (Sawaf) (Karlbom, 2016) (Thurmair). The most common forms of hybrid MT are Multi Engine, Statistical rule generation and multi pass. RBMT is a linguistic approach and SMT is probabilistic approach, combining both approach results in quite efficient output (Simard, 2007.) (Groves, 2005) (Dugast, 2007).

In Table 1, we have compared various existing MTS with their approach and accuracy.

III. RELATED WORK

In this section, various research work carried out for processing Sanskrit language and Hybrid MTS formed have been discussed.

In 1987, an attempt to parse the Sanskrit sentence mechanically by Pushpak Bhattacharya in his M.tech thesis at IIT, Kanpur (Bhattacharyya, 2009). Another individual attempt in 1990's includes verbal cognition generator for bhandarkar's Sanskrit primer by Pr. Lakshmitatachar at Academy of Sanskrit research in Melkote (Ramapriya, 2001). After this several other Sanskrit dictionaries were digitized. The Apte's and MacDonell's Sanskrit-English dictionaries digitized among Indian language dictionaries at university of Chicago under the digital dictionaries of South Asia project. In1994, Gerard Huet started developing Sanskrit Heritage site which includes Sanskrit heritage dictionary (Sanskrit – French), which served as a morphology generator and was equipped with grammatical tools and later Monier-William by Thomas Malten's at koln University digitalization was added to the site (Digital Dictionaries of South Asia project.). The website offered various Sanskrit linguistic services such as Sanskrit reader which parses the Sanskrit text under various formats into Sanskrit banks of tagged hypertext. Various other tools such as Sanskrit Parser, Lemmatiser, Tagger, Morphological and phonological tools are provided by the website (Site.). It was later equipped with Index program and audio features in 2001 by Hyman. It helped Ramopakhyana to be searched by inflectional and lexical categories along with verbal roots, text ranges, nominal stems etc. Various tools for morpho-phonemic computation and lexical representation were adapted from Zen library, a general computational linguistic toolkit which implements finite state transducers in functional programming language (Scharf, 2002). It was an instance of new relational computing paradigm called effective Eilenberg machines (Razet, 2008) (Huet, 2002.). In 2006-2009, at Brown University's Classical Department an International Digital Sanskrit Library Integration Project which integrated all digital Sanskrit libraries (Monier Williams and TITUS dictionaries). This integrated dictionary was improved by adding tagging information, systematically classifying and converting code markers to explicit XML tags by JIM Funderburk and R. Chandrashekar.

Table 1. Hybrid Machine Translation System

Machine translation Systems	**Approach**	**Language pair**	**Accuracy**
Anuvaadak machine translation system (Kunchukuttan *et al.*, 2014)	Hybrid approach (SMT and transliteration)	English-Hindi	BLEU score for Indo-Aryan languages is 35.73% and Dravidian is 6.75%.
A hybrid approach to English to Malayalam machine translation (Nithya, 2013).	Hybrid approach (SMT and TM (Translation memory)	Malayalam-English	BLEU score for the baseline system was 68.14 and for the hybrid system was 69.33
Translation for UI labels of commercial web based interactive applications (Dhore and Dixit, 2011)	Hybrid based (Rule based and transliteration)	English-Devanagiri	The system observed correct matches 95% for English-Hindi language pair, 95.5% for English to Marathi and 96.5 % for English to Gujarati.
Lattice based lexical transfer (Chatterji, 2011)	Hybrid based (SMT with lattice and synonyms choices)	Bengali-Hindi	BLEU score of 25% for the baseline system and 29% for the modified system
Hindi to Punjabi machine translation system (Goyal and Lehal,2011)	Hybrid based (Direct and rule-based system)	Hindi-Punjabi	Accuracy achieved by the system is 87.60%.
Translation rules and ANN based model (Khan and Mishra, 2011)	Artificial neural network and rule-based approach.	English-Urdu	n-gram BLUE score is 0.6954, METEOR score is 0.8583 and F-score is 0.8650
Sampark()	Hybrid approach (Statistical and rule based)	Indian languages to Indian languages	In intelligibility test accuracy is 70% and for accuracy test is 50%.
Developing English-Urdu machine translation (Sinha and Mahesh)	Interlingua and rule-based approach.	English-Urdu	BLEU score for the system is 0.3412 for Hindi and 0.3544 for Urdu.
Bengali to Hindi Machine Translation System (Chatterji *et al.*, 2009)	Hybrid (statistical and rule based)	Bengali –Hindi	BLEU score of SMT system is 0.1745 and lexical transfer-based system is 0.424 and hybrid system is 0.2275
Anubharti (Sinha, 2014)	Hybrid approach (Example and rule based)	Hindi –English	It works accurately for noun and verb phrases.

The Vedic text was represented digitally named standard version 5.2. Devanagari extended and Vedic extensions under south Asian script were also made available online (Project.). In 2010, Sanskrit Library and Sanskrit heritage site were linked. Sentences where sandhi has not been analysed is directed to Sanskrit Heritage parser which is a shallow parser the parser recognizes sentence using syntactic criterion and full-form lexicon of 70,000 form derived from about 25,000 words of SH lexicon. Amba Kulkarni's dependency parser at University of Hyderabad also has been linked to Sanskrit Heritage site (Kulkarni, 2010) (Kulkarni A. a., 2011). There are various MTS developed for Indian language pairs using hybrid approach are compared in Table:1 with their accuracies or outcomes.

IV. PROPOSED METHODOLOGY

In this section, a novel approach has been proposed to develop hybrid MTS model as presented in Figure 2 (Singh *et al.*, 2019). In the existing model, the rule-based machine translation system is used by following three basic steps: Analysis, transfer and generation. When input text is passed to MT system, it is processed by the number of processes such as morphological analyser, POS tagger, Lexical selection, structure transfer, morphological generator and post-generator gives target or output text. Here the first step includes data analysis, second includes transfer the data in

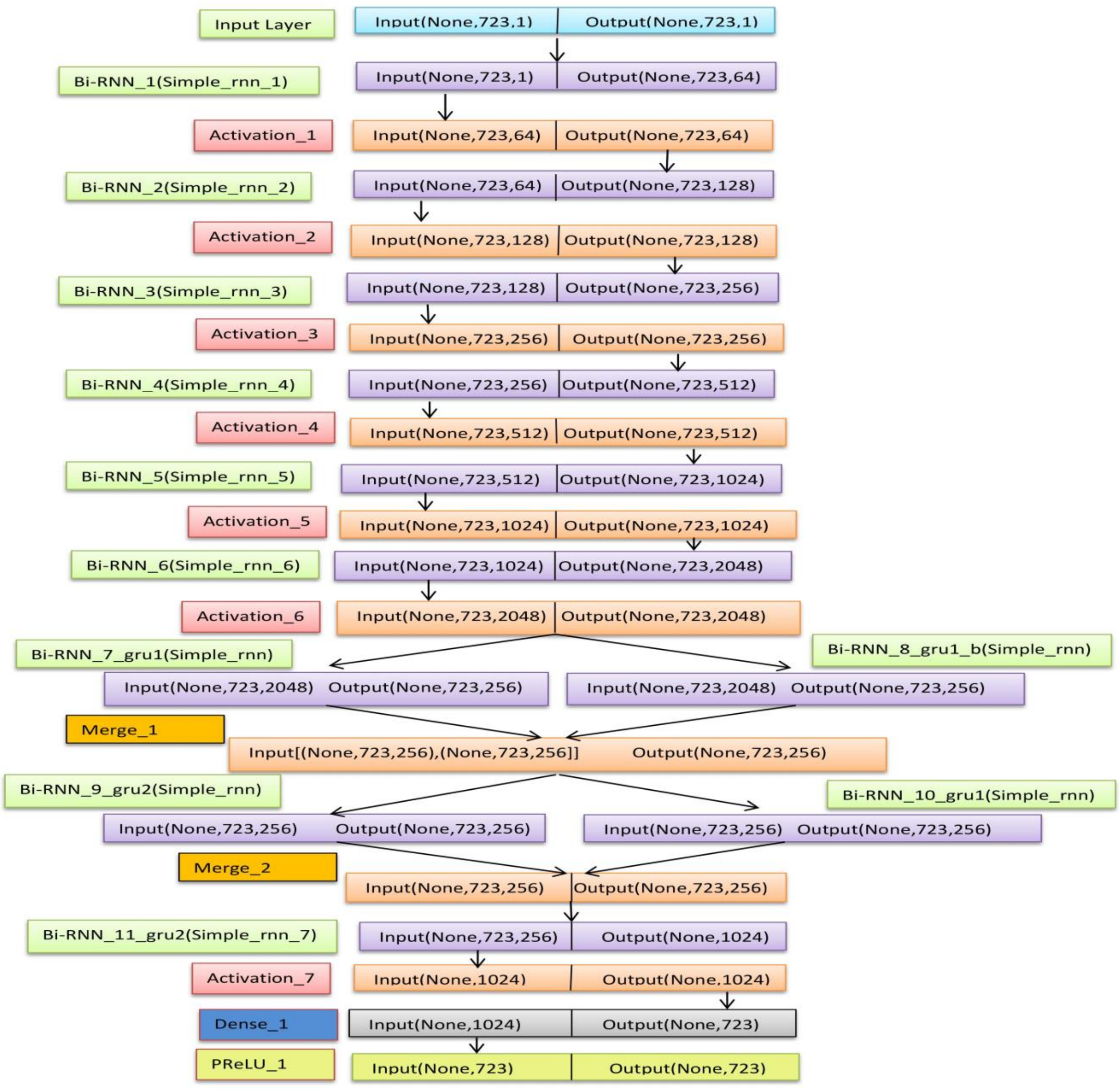

Figure 1. Flow of Neural Network Model

which source language parser representation is converted to the target language and the last steps include the text generation to generate final target language. In the existing Sanskrit based system, there are various drawbacks such as slow speed, less data accuracy, low response time and time-consuming. All these factors are responsible to degrade the performance of the system. Therefore, a new model is required to overcome these setbacks. A major issue which arises in implementation of Sanskrit based machine translation system is the approach for developing system In this paper, a new hybrid model is proposed using deep learning which gives accurate results as compared to the existing model. In this section, we will discuss the mathematical model of the proposed system, its flowchart and algorithms respectively.

In the proposed model, no rule-based approach is used. Here huge dataset is trained using a tensor flow model and will pass data through some specific number of hidden layers which will make this proposed model accurate. This hybrid model will reduce the complexity of existing CGI-model and will make the system faster and accurate.

In the proposed hybrid model, train this proposed model using both the rule-based approach and the neural network model. Here the input is passed through both the approaches where first it follows a rule-based approach and then passed to a proposed hybrid model with a huge number of hidden layers that layers will be added automatically through auto-tuning. The auto-tuning has helped to make this model better than all other previous models. Now, the system will automatically calculate the required number of input layers which makes the proposed hybrid model accurate. And then it is trained using a large number of hidden layers at a very high speed which makes it better among all.

V. MATHEMATICAL MODEL

Given a source sentence $x = x_1 \dots \dots x_n$ and a target sentence $y = y_1, \dots \dots . y_m$, the model first tokenise x to form input representations where tokenization is the first step where the probability of a sequence of T words is denoted as $P1 = W_1 \dots \dots \dots . . W_T$. Since the number of words coming before a previous word w1 varies depending on locations with input document is usually conditioned on a window of words rather than all previous words.

$$P_1(W_1 \dots \dots . W_T) = \prod_{i=1}^{i=T} P(W_i | W_1 \dots . W_{i-1})$$

$$\approx \prod_{I=1}^{i=T} P(W_i | W_1 \dots , W_{n-1}), . . W_{i-1}) \quad (2)$$

Equation 1 is used for tokenization. Then tokenised data is passed through the encoder network that takes the input sequence and maps it to an encoded representation of the sequence. The encoded representation is then used by the decoder network to generate an output sequence. Equation 2 shows the relationship for the encoder stage. Equation 3 and 4 show the equation of the decoder stage.

$$h_t = \varphi(h_{t-1}, x_t) = f\left(W^{(hh)} h_{t-1} + W^{(hx)} x_t\right) \quad (3)$$

$$h_t = \varphi(h_{t-1}) = f\left(W^{(hh)}\, h_{t-1}\right) \quad (4)$$

$$y_t = softmax(W^{(S)} h_t) \quad (5)$$

To predict the target labels y, the model used Learning Classifier for classification. The objective of training classifier is to minimize the number of errors. If $f : R^D \to \{0 \dots \dots . L\}$ is the prediction function and loss can be calculated as: $L_{0,1} = \sum_{i=0}^{101} I_f (x^{(i)}) \neq y^{(i)})$, D is the training set or $D \cap D_{train} = \phi$ (to avoid the evaluation of Validation or test error. I is the indicator function defined as: I_x= 1 if x is true otherwise is defined as

$$f(x) = argmax_k P(y = k | x, \theta) \quad (6)$$

In classification, each timestamp, the network maintains two hidden layers, one for the left to right propagation and another for the right to left propagation. Then 5 and 6 equations show the mathematical function behind setting up the bidirectional RNN hidden layer. The only difference between these two relationships is in the direction of recurring through the corpus the equation 7 shows via summarizing padding and future word representation.

$$\overrightarrow{h_t} = \mathrm{f}(\overrightarrow{w_{xt}} + \overrightarrow{V}\, \overrightarrow{h_{t-1}} + \overrightarrow{b}\,) \quad (7)$$

$$\overleftarrow{h_t} = f(\overleftarrow{W_{xt}} + \overleftarrow{V}\overleftarrow{h_{t+1}} + \overleftarrow{b} \quad (8)$$

$$y_t = g(uh_t + C) = g\left(u[\overrightarrow{h_t}; \overleftarrow{h_t}] + c\right) \quad (9)$$

Further, Gated Recurrent units are designed in a manner to have more persistent memory thereby making it easier for RNN capture long term dependencies. Mathematically in Eq (10) GRU has $h^{(t-1)}$ and $x^{(1)}$ to generate the next hidden state $h^{(1)}$

$$z^{(t)} = \sigma\left(W^{(Z)}x^{(t)} + u^{(Z)}h^{(t-1)}\right) \qquad (10)$$

For update gate Eq.(10), for reset gate in Eq. (11) , new memory in Eq. (12) and hidden state in Eq. (13).

$$r^{(t)} = \sigma(W^{(r)}x^{(t)} + u^{(r)}h^{(t-1)} \qquad (11)$$

$$\overline{h^{(t)}} = \tanh(r^{(t)} \circ U_h\ h^{(t-1)} + w_x^{(t)} \qquad (12)$$

$$h^{(t)} = \left(1 - z^{(t)}\right) \circ \overline{h^{(t)}} + z^{(t)} \circ h^{(t-1)} \qquad (13)$$

To predict target labels more efficiently activation function is being passed to the model. It also works as a rectifier for us. Here we are using Relu and sigmoid activation function. These activations are mainly non-linear activations which results in faster training of neural networks. Then this input is passed for activation as in Eq. (14)

$$Z = b + \sum_i w_i x_i \qquad (14)$$

For the Sigmoid activation function, input will be as given in Eq. (15) and for Relu activation function input will be as shown in Eq. (16), whereas its derivative is represented in Eq. (17).

$$\sigma(X) = \frac{1}{1+e^{-x}} \qquad (15)$$

$$f(\alpha, x) = \left\{\begin{pmatrix} \alpha x\ for & x < 0 \\ x\ for & x \geq 0 \end{pmatrix}\right\} \qquad (16)$$

$$f(\alpha, x) = \begin{pmatrix} \alpha\ for & x < 0 \\ 1\ for & x \geq 0 \end{pmatrix} \qquad (17)$$

Having range (-∞, ∞). Finally, model gives the target value, then it will be passed to hidden decoder state, to make output to human readable form, then label y_t is predicted using a learned distribution $p(y_t\ |h_t\ )$.

A. Algorithms

In this section, the data flow diagram and algorithms for the proposed hybrid model are discussed. These algorithms are categorized into two parts i.e. data Tokenization and model development. After data gathering which includes the collection of all data and separates the meaningful data from it. All the meaningful data is gathered in a huge file in a required pattern. After that, data tokenization includes the part in which data gets tokenized which means we split all the sentences into words, remove all the symbols, tags or any other redundant words from it. It also includes the encoding part of data which means we encode the data into binary format to make it secure and machine understandable. At last, model development includes the splitting of data into training and testing of data and then passing the data to the model. Here we build and save the model by passing the data to it. DFD (as shown in Figure 1) and Algorithms of the proposed model.

This step includes the collection of the dataset containing millions of word meanings in a systematic way from the different resources to make it useful for the proposed model. From all sparse spreader, datasheets make big dataset and save it in a proper word meaning format. If the dataset is not in the correct format, marked it as a redundant dataset. After collecting all the dataset in a single CSV, clean the dataset by removing all the redundant or unused words and extra symbols present in the dataset. Here we arrange the dataset in a proper structure to make it suitable for our model. After that, visualize the dataset to check the structure and the correlation in the dataset. If there is no relation found in the dataset then we do pre-process of data again from starting. Tokenization is the chopping of data into words. In this step store all the dataset in a matrix format and then Tokenize the data by splitting all the sentences into words and label the data into numeric form. As this data is in Sanskrit and Hindi language, encode it normally as done for English words So here use Count vectorise. Count-Vectorise will convert all type of data to numeric form easily so use it for labelling the data.

After spitting the encoded data into training and testing sets, we train the model by passing the training set into it and add some hidden bidirectional layers to proposed hybrid model. At the end, we attain 99.9% accurate model. After tokenization data is analysed through lexical and semantic analysis. Here arrange the data in a proper grammatical structure with respect to the Sanskrit rules.

Algorithm 2: Model Development

```
1. Get the encoded data (data).
2. Split the data into training and testing form
        X = data[:,x:y]
        Y = data[:,z]
   Where x,y,z are number of rows and columns
   we used to train the model
3. Create the sequential model using keras.
        model = Sequential()
4. Add hidden layers with different activations i.e.
   model.add (
                Dense(12, input_dim=723,
                init='uniform',
                activation='relu'))
                model.add(Dense(8,
                init='uniform',
                activation='relu'))
                model.add(Dense(1,
                init='uniform',
                activation='sigmoid')
        )
5. Train and test the model.
6. while (accuracy < 95%):
        switch(i):
                case 1:
                add bidirectional layer
                add gru activation to it
                i++
                case 2:
                add second bidirectional
                layer
                add gru activation to it
                i++
                case 3:
                add third bidirectional layer
                add gru activation to it
                i++
                case 4:
                add fourth bidirectional layer
                add activation to it
                i++
                case 5:
                add fifth hidden layer
                add Relu activation to it
                i++
                case 6:
                add Dense layer
                add linear activation to it
```

Algorithm 1: Data Tokenization

```
1. Read data from csv
2. Convert the data in matrix form using

        Matrix data = data.asmatrix( )

3. Pass the matrixdata to count Vectorizer
   using

   vect = CountVectorizer()
   vect = CountVectorizer(
                stop_words='english',
                ngram_range= (1, 1),
                dtype='double' )
   data =vect.fit_transform(matrixdata)

4. Check the vocabulary to prevent redundancy
   using:

        countvectorizer.vocabulary

5. If (redundant word)

   {
        Ignore the word or label it as 0.
   }
   Else
   {
   Label the data in numeric form.
   }
```

To overcome the problem of overfitting and under fitting, we split the data in two parts out of which one is used for the model development for predictive analysis and the other one is used for performance analysis. Divide the data into two parts for training and testing purpose. Now pass the training set into model. to second layer RNN. Input layer is the layer which interacts with hidden layers. Our accuracy depends on the number of input layers we passed and the number of times it interacts with the hidden layers. Here, Input is passed through the first layer of recurrent networks (RNN) and the number of neurons. Secondly, we applied activation functions, and which gives the probability of output. In the end, it passes the output of first RNN More the interaction between the input layers and hidden layers more is the accuracy attained and vice versa. Now, first of all, normalize the batch and again apply the activation function. Passes output of second RNN to third layer RNN. Again, normalizing the batch and apply the activation function. Then Passes output of Third RNN to fourth layer RNN. Repeat this process up to seventh layer RNN. After the addition of layers, divide and merge the datasets according to as per the requirement. Normalize the batch again. Again, applied activation functions and gives the probability of output. Again, divide and merge the data as per requirement neurons are passed in dense step. Then the output is generated at the end that exhibits the accuracy of 99.97%.

VI. EXPERIMENTAL RESULTS

In this section, the evaluation of experimental results has been discussed. The proposed Neural network machine translation system consists of various steps through which the proposed model is trained and processed.

A. Technical Details

The sections contain detail of setup, dataset and training details covered in detail.

1. Setup

In the proposed work, Keras sequential model is used to process the data. The proposed model is processed through highly configured core GPU with 32 GB of RAM to achieve a high throughput speed approximately 2500 words per second. This speed is not possible for normal systems because in this one epoch will take approximately two hours to run. So we use a highly configured GPU along with NVIDIA Geforce GTX 980 GPU.

2. Dataset Details

For the proposed model huge amount of dataset is collected which consists of Sanskrit to Hindi word meanings as shown in Table 2. Along with the word meanings data, it also consists of various rules of Sanskrit as well as Hindi. Thus, redundant data from that dataset has been removed and arranged in proper word meaning format to pass it through the model. Also, collect the huge dataset in a single csv file which contains around millions of sentences.

3. Data Pre-Processing

The collected data was non-uniform and unformatted data. Therefore, various ways are needed to make it suitable for the proposed model. The single CSV file is processed twice for rechecking to remove redundant words. If any redundant words are found it removes them from there otherwise it does encode of the dataset.

4. Training Details

Data is trained using the model described above i.e. Keras sequential model with the TensorFlow at the backend. When the dataset is passed only through the Keras model it is not accurate. To achieve accuracy, add some hidden layers to it through which accuracy is improved but it is not sufficient. Then for further improvements add some activation functions to it and extracts the useful information and removes extra noises. As a result, an accurate model is achieved. For making the proposed model best among all, auto-tuning is also added which automatically adds the required number of input layers before training according to data passed. After that proposed model will become the best, which is very fast, highly précised and accurate.

VII. EVALUATION METRICS

We evaluate hybrid model using various evaluation measures described in detail in further sections.

A. BLEU

BLEU score is an important metric used for calculating the accuracy of translated sentences as compared to the human-generated reference translations. It is not good for shorter translations, but it provides accurate results for longer sentences. Normally Bleu Score values lie between 0 and 1, simply multiplying it to 100, its percentage can be calculated. It is observed that the higher the bleu score value, the model is more accurate. The formula of BLEU Score is shown in Eq. (18) (Lavie, 2016).

$$\text{BLEU}= \min(1,\frac{Output-length}{reference-length})(\prod_{i=1}^{4} precision_i) \qquad (18)$$

BLEU score of a different model is shown in Table 3. BLEU score is widely used for machine translation systems. It is used to calculate the score of the accuracy of the translated sentence as compared to the human-generated translations. It will compute precision w.r.t human-generated translation without taking into account any grammatical corrections/errors. Its performance is not accurate if used to judge individual sentence or for shorter translations. Formally it lies between 0 and 1 and calculation in percentage also by simply multiplying it to 100. Higher the value of BLEU score, better the model. In Table 3, there is a comparison of various models in terms of BLEU score. When data is trained using simple Keras model, the BLEU score is 10.23% which is too low. Then add Bi-directional layers to the model. Then it gives the BLEU score of 29.12% which is also less. Then add Gru layers along with it and got BLEU to score 40.32%. To improve it further add ReLu and sigmoid activation functions to it and got 56.78% BLEU score which is quite good, but in the hybrid model when the model is trained along with auto-tuning it gives BLEU score 61.02% which is very good. It proves that the hybrid model is better amongst all. In Figure 11, it is observed that the Hybrid model has higher Bleu Score value. In the proposed model, we have used Back Propagation along with various hidden layers, which make it better than other existing models.

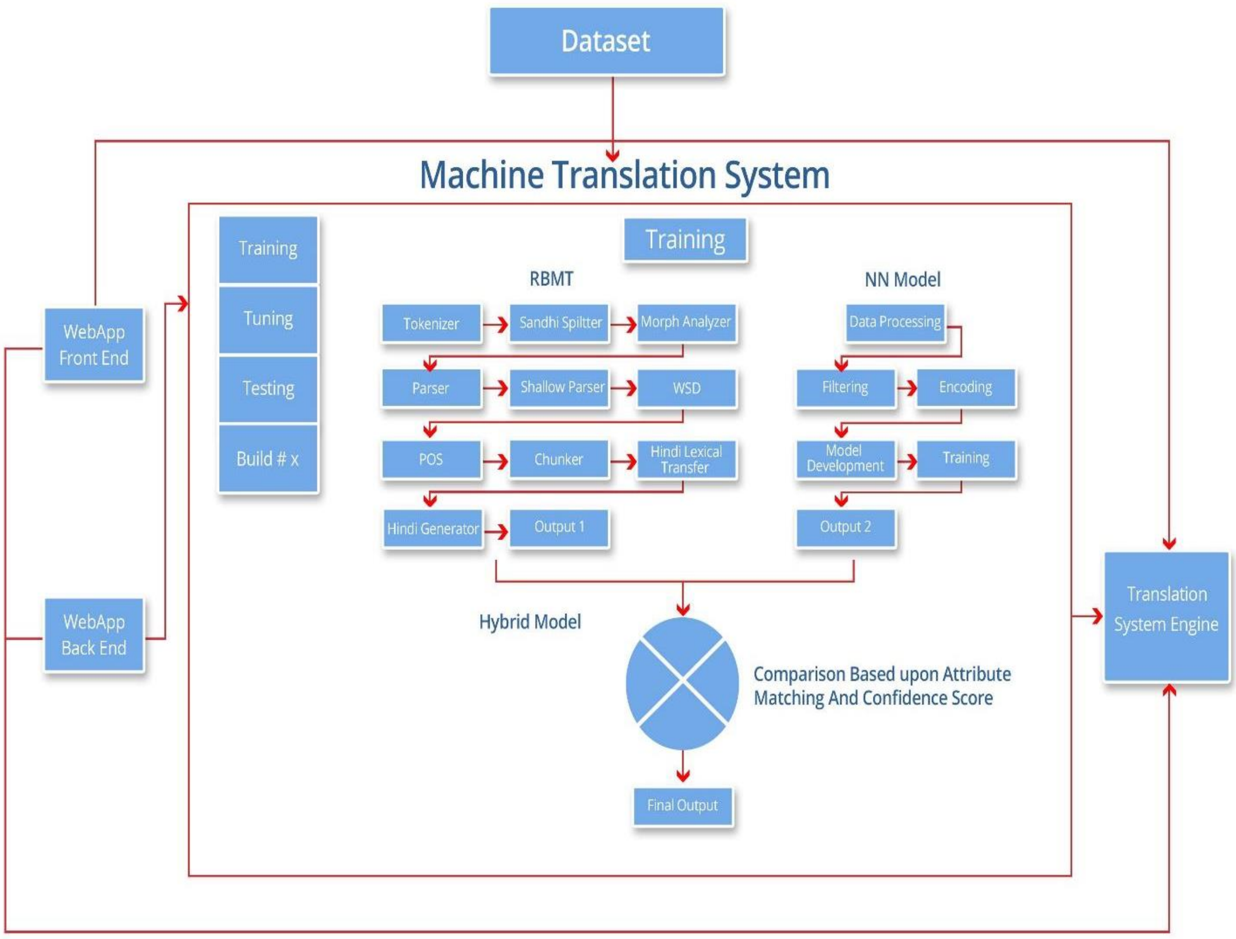

Figure 2. Proposed Architecture of Machine Translation System

Table 2. Dataset

Dataset	**#Sentences**	**#Words**		**Vocabulary**	
		Source	**Target**	**Source**	**Target**
Corpus: Open MT Sanskrit-English					
Training	14,534,215	131,575,835	123,425,654	355.465	124.278
Development	192,679	172,799	122,645	NA	NA
Test	12,698	77,322	26,273	NA	NA
Corpus: Open BTEC English-Hindi					
Training	12,643,124	11,425,429	11,753,927	428.672	111.249
Development	13,679	122,769	87,286	NA	NA
Test	10,684	67,827	39,207	NA	NA

Table 3. Models with iterative increment

Models	**BLEU Score (in %)**
Simple Sequential Model	10.23
Keras Model with bi-directional layers	29.12
Keras Model + bi-directional layer + gru	40.34
Keras Model + bi-directional layer + gru+ Relu and sigmoid activation function	56.78
Keras Model + bi-directional layer + gru+ Relu and sigmoid activation function + auto-tuning	61.02

B. WER

WER stands for Word error rate. It is a metric used to calculate the error rate by comparing machine translated output with the human translated output.

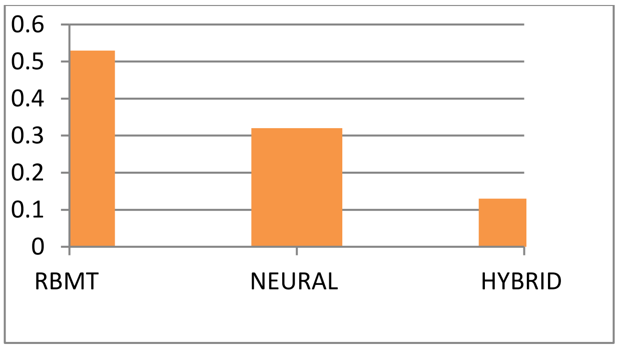

Figure 3. WER Evaluation of RBMT, Neural and Hybrid

$$\text{WER}=\frac{substitutions+insertions+deletions}{reference-length} \tag{19}$$

WER is calculated with Eq. (19). Here substitution means replacement of one word with another. Insertion means the addition of words and deletion means dropping of words. Figure 3, WER of various models such as RBMT, Neural and Hybrid are discussed. It is observed that the Hybrid model has less WER as compared to other existing models. The less WER, the more accurate the model. BLEU Score and WER are inversely proportional to each other. It implies that if BLEU score is more than WER will be less and vice versa.

C. F-Measure

It is a metric used to calculate the accuracy and precision of the model. It is used to calculate the quality or exactness of an output. Higher the F-Measure, better the system. For the F-measure, we need precision and recall values also.

$$\text{Precision}=\frac{Correct}{Output-Length} \quad (20)$$

$$\text{Recall}=\frac{Correct}{reference-length} \quad (21)$$

$$\text{F-measure}=\frac{Precision \times recall}{(Precision+\Re call)/2} \quad (22)$$

In Figure 5, F-measure of various models such as RBMT, Neural and Hybrid are discussed. It is observed that the Hybrid model has more F-measure as compared to other existing models. It is observed that if more the F-measure more will be accuracy.

D. METEOR

It is used to find the correlation between the machines translated output and the human-generated sample output. It is assumed that Higher the Meteor value, better the model will be. The METEOR Score, for the given elements, can be calculated as in Eq. (23).

$$\text{Score}= \text{Fmean}*(1\text{-penalty}) \quad (23)$$

$$\text{Fmean}=\frac{10PR}{(9+RP)} \quad (24)$$

$$\text{Penalty}=0.5\left(\frac{chunks}{unigrams_{matched}}\right)^3 \quad (25)$$

It is helpful to reduce the effect of Fmean. To calculate Fmean, formula is (T. Mikolov, 2010). Figure 4 depicts METEOR of RBMT, NEURAL and Hybrid. For longer values, the penalty is needed to be calculated. Thus, Penalty is

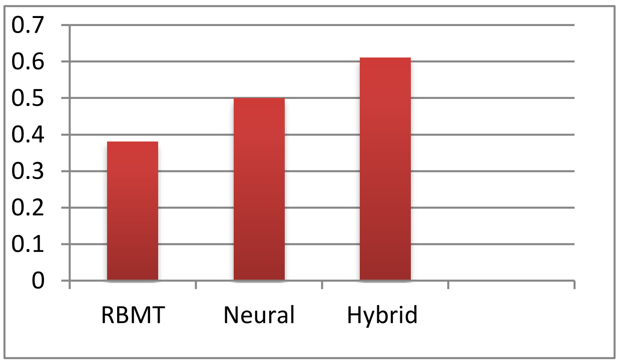

Figure 5. F-Measure of RBMT, Neural and Hybrid

calculated as in Eq. (25). In the proposed Hybrid model, Meteor value is high as compared to other models due to the high correlation between the words of the output sentences.

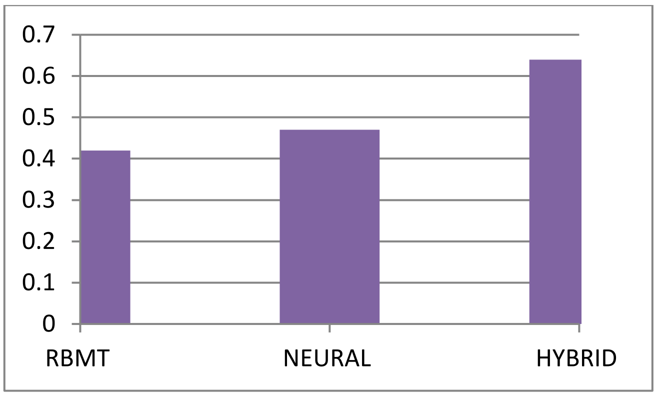

Figure 4. METEOR evaluation of RBMT, NEURAL and HYBRID model

VIII. RESULTS AND DISCUSSIONS

The BLEU metric parameters and human evaluation metrics of adequacy, fluency and relative ranking values were used to evaluate the performance of the models. Table 5 describes the comparison of the accuracy of Neural and hybrid neural model. When the data is passed from the tensor flow model only, less accuracy and speed is attained. When combined, both the models it will attain maximum accuracy and will start processing the input and generates output faster than the previous model.

Table 4. Comparison of Model with their accuracy

Sr. No	**Model**	**Accuracy**
1.	Neural Model	99.97%
2.	Hybrid model (Neural and RBMT)	99.99%

When the data is passed from tensor flow model only, less accuracy and speed is attained. When combined, both the models it will attain maximum accuracy and will start Here P is Precision and R is recall. Mean, Precision and Recall are based upon the unigrams matches. An error analysis based on linguistic types of sentences.

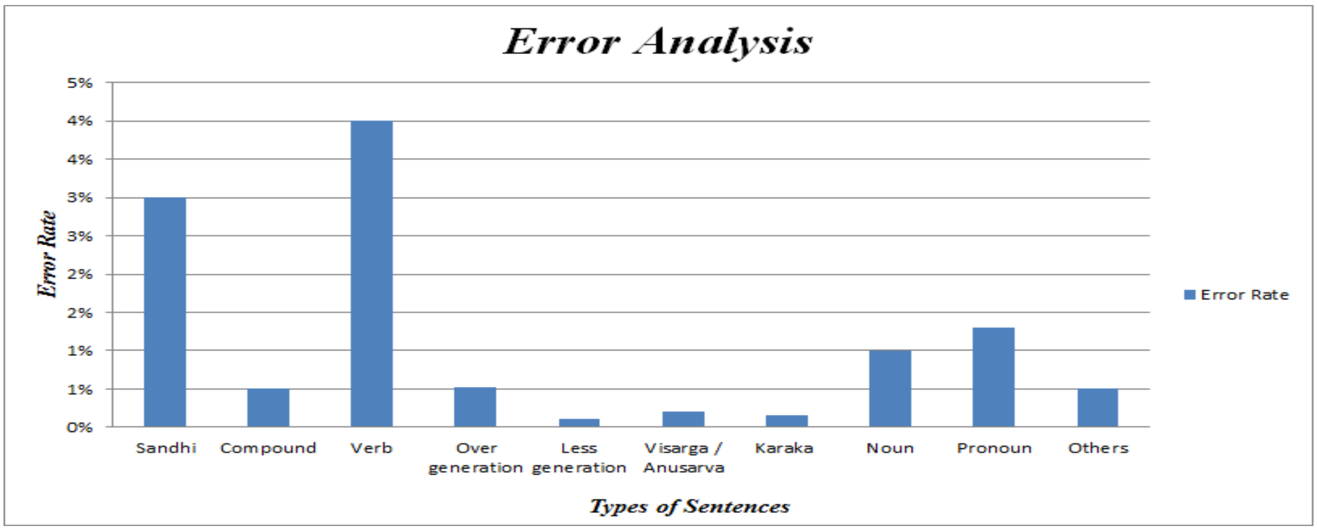

Figure 6. Case study on type of sentences

IX. CONCLUSION

Sanskrit to Hindi Translation is one of the most challenging tasks. As a result, the model became complex and time-consuming. In this paper to overcome the existing problem, deep learning has been implemented to train the data and numpy, pandas and sklearn libraries for model building. In proposed work, Keras is used as a front end and Tensor flow is used as a back-end library. Performance evaluation showed that the proposed MT system gives better performance in terms of accuracy, speed and response time than [23]. As a result, we have attained 99.99% model accuracy and a BLEU score of 61% which is higher than RBMT which is 41%. The proposed hybrid model is fast and more efficient than the existing RBMT. This proposed hybrid model is more efficient in terms of non-retrieval of text. If in any case rule-based approach becomes infeasible to return output, then it returns the output using the direct approach. In non-rule match cases, the rule-based model does not return any output, on the other hand, our proposed model always returns the best solution. The complexity of existing RBMT system becomes very high for long sentences and these are practically infeasible sometimes, but the proposed model is efficient for such cases also as shown by us in results. In future, multiple linguistic languages can be taken to convert into the single target language and multi-lingual platform can be used for this purpose. Even accuracy of the system would be more improved by addressing various problems which would come in future.

X. REFERENCES

Banerjee, S. and Lavie, A., 2005, June. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (pp. 65-72).

Bharati, A., Chaitanya, V., Sangal, R. and Ramakrishnamacharyulu, K.V., 1995. *Natural language processing: a Paninian perspective* (pp. 65-106). New Delhi: Prentice-Hall of India.

Chatterji, S., Roy, D., Sarkar, S. and Basu, A., 2009. A hybrid approach for bengali to hindi machine translation. In *Proceedings of ICON-2009: 7th International Conference on Natural Language Processing* (pp. 81-91).

Chatterji, S., Sonare, P., Sarkar, S. and Basu, A., 2011. Lattice Based Lexical Transfer in Bengali Hindi Machine Translation Framework. In *Proceedings of ICON-2011: 9th International Conference on Natural Language Processing*.

Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Dhore, M.L. and Dixit, S.X., 2011. English to Devnagari translation for UI labels of commercial web based interactive applications. *International Journal of Computer Applications*, *35*(10), pp.0975-8887.

Digital Dictionaries of South Asia project. *Available online at: http://dsal.uchicago.edu/dictionaries/list.html*.

Dorr, B.J., Hovy, E.H. and Levin, L.S., 2004. Machine translation: Interlingual methods.

Dugast, L., Senellart, J. and Koehn, P., 2007, June. Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 220-223).

Fogel, D.B., 2001. *Blondie24: Playing at the Edge of AI*. Elsevier.

Forcada, M.L. and Ñeco, R.P., 1997, June. Recursive hetero-associative memories for translation. In *International Work-Conference on Artificial Neural Networks* (pp. 453-462). Springer, Berlin, Heidelberg.

Goyal, P., Huet, G., Kulkarni, A., Scharf, P. and Bunker, R., 2012, December. A distributed platform for Sanskrit processing. In *Proceedings of COLING 2012* (pp. 1011-1028).

Goyal, V. and Lehal, G.S., 2011, June. Hindi to Punjabi machine translation system. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations* (pp. 1-6). Association for Computational Linguistics.

Groves, D. and Way, A., 2005, June. Hybrid example-based SMT: the best of both worlds?. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts* (pp. 183-190). Association for Computational Linguistics.

Karlbom, H., 2016. Hybrid Machine Translation: Choosing the best translation with Support Vector Machines.

Khan, S. and Mishra, R.B., 2011. Translation rules and ANN based model for English to Urdu machine translation. *INFOCOMP*, *10*(3), pp.36-47.

Koehn, P., Och, F.J. and Marcu, D., 2003, May. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 48-54). Association for Computational Linguistics.

Kulkarni, A. and Kumar, A., 2011. Statistical constituency parser for Sanskrit compounds. *Proceedings of ICON*.

Kulkarni, A., Pokar, S. and Shukl, D., 2010, December. Designing a constraint-based parser for Sanskrit. In *International Sanskrit Computational Linguistics Symposium* (pp. 70-90). Springer, Berlin, Heidelberg.

Kulkarni, M. and Bhattacharyya, P., 2007. Verbal roots in the Sanskrit WordNet. In *Sanskrit Computational Linguistics* (pp. 328-338). Springer, Berlin, Heidelberg.

Kulkarni, M. and Bhattacharyya, P., 2007. Verbal roots in the Sanskrit WordNet. In *Sanskrit Computational Linguistics* (pp. 328-338). Springer, Berlin, Heidelberg.

Kunchukuttan, A., Mishra, A., Chatterjee, R., Shah, R. and Bhattacharyya, P., 2014. Sata-anuvadak: Tackling multiway translation of indian languages. *pan*, *841*(54,570), pp.4-135.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J. and Khudanpur, S., 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Nithya, B. and Joseph, S., 2013. A hybrid approach to english to malayalam machine translation. *International Journal of Computer Applications*, *81*(8).

Noone, G., 2003. Machine Translation A Transfer Approach. *Computer Science, Linguistics and a Language (CSLL) Department, University of Dublin, Trinity College, Final Rep*.

Project., Available online at *sanskritlibrary.org/nsf2005.html*. Accessed on: 30-3-2019

Razet, B., 2008, July. Finite Eilenberg machines. In *International Conference on Implementation and Application of Automata* (pp. 242-251). Springer, Berlin, Heidelberg.

Sampark: Machine Translation System among Indian languages (2009) http://tdildc. in/index.php?option=com_vertical&parentid=74, http://sampark.iiit.ac.in/

Sawaf, H., Shihadah, M. and Yaghi, M., eBay Inc, 2017. *Hybrid machine translation*. U.S. Patent 9,798,720.

Scharf, P.M., 2003. *Rāmopākhyāna: the story of Rāma in the Mahābhārata: an independent-study reader in Sanskrit*. Psychology Press.

Simard, M., Ueffing, N., Isabelle, P. and Kuhn, R., 2007, June. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 203-206). Association for Computational Linguistics.

Singh, M.,Kumar R., and Chana,I., 2019.Improving Neural Machine Translation using Rule-based Machine Translation. In 7th International Conference on Smart Computing and Communications (ICSCC-2019), IEEE .

Sinha, R.M.K. and Mahesh, K., 2009, August. Developing english-urdu machine translation via hindi. In *Third Workshop on Computational Approaches to Arabic-Script-based Languages*.

Sinha, R.M.K., 2004, November. An engineering perspective of machine translation: anglabharti-II and anubharti-II architectures. In *Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS-2004)* (pp. 10-17).

Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).

Thurmair, G., Will, T. and Aleksic, V., Linguatec Sprachtechnologien GmbH, 2008. *Hybrid Machine Translation System*. U.S. Patent Application 11/885,688.

www.statmt.org. (n.d.). *http://www.statmt.org/book/slides/08-evaluation.pdf*. Accessed on: 30-2-2019

www.statmt.org. (n.d.). http://www.statmt.org/book/slides/08-evaluation.pdf. Accessed on: 30-2-2019

www.wildmil.com. Accessed on: 30-2-2019.