

# Sanskrit to Hindi language translation using multimodal neural machine translation

Prashanth Kammar<sup>1</sup>, Parashuram Baraki<sup>2</sup>, Sunil Kumar Ganganayaka<sup>3</sup>, Manjunath Swamy  
Byranahalli Eraiah<sup>4</sup>, Kolakaluri Lakshman Arun Kumar<sup>5</sup>

<sup>1</sup>Department of Computer Science and Engineering, Proudhadevaraya Institute of Technology, Hosapete and Visvesvaraya Technological University, Belagavi, India

<sup>2</sup>Department of Computer Science and Engineering, Smt. Kamala and Sri Venkappa M Agadi College of Engineering and Technology, Lakshmeshwara and Visvesvaraya Technological University, Belagavi, India

<sup>3</sup>Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bengaluru, India

<sup>4</sup>Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru, and Visvesvaraya Technical University, Belagavi, India

<sup>5</sup>Department of Computer Science and Engineering, KNS Institute of Technology, Bengaluru, and Visvesvaraya Technical University, Belagavi, India

## Article Info

### Article history:

Received Jan 19, 2024

Revised Feb 18, 2024

Accepted Feb 20, 2024

### Keywords:

Attention mechanism

Decoder

Encoder

Neural machine translation

Semantic text

## ABSTRACT

Machine translation (MT) is a subfield of computer features that focuses on the automatic translation from one natural language into another without any human involvement. Due to native people interacting in a variety of languages, there is a great need for translating information between languages to send and communicate thoughts. However, they disregard the significance of semantic data encoded in the text features. In this paper, multimodal neural machine translation (MNMT) is proposed for Sanskrit-Hindi translation. The main goal of the proposed method is to fully utilize semantic text features on NMT architecture and to minimize testing and training time. The MNMT is validated on two different NMT architectures: recurrent neural network (RNN) and self-attention network (SAN). The MNMT method's efficacy is demonstrated by employing the dataset of Sanskrit-Hindi Corpora. Extensive experimental outcomes represent the proposed method's enhancement over baselines on both architectures. The existing methods, namely, English-to-Indian MT system, Sanskrit-Hindi MT system, and hybrid MT system are used to justify the efficacy of the MNMT method. When compared to the above-mentioned existing methods, RA-RNN respectively achieves a superior BLEU and METEOR of 80.5% and 75.3%, while the RA-SAN respectively achieves a superior BLEU and METEOR of 78.2% and 77.1%.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Prashanth Kammar

Department of Computer Science and Engineering

Proudhadevaraya Institute of Technology, Visvesvaraya Technological University

Belagavi, India

Email: prashanthkogali@gmail.com

## 1. INTRODUCTION

Machine translation (MT) is one of the earliest and most interesting fields of natural language processing (NLP). MT is a subfield of computer features that focus on the automatic translation from one natural language into another, without any human involvement [1]. Indian language's text reflects variations of rich morphology and different word orders in their representation required to be considered by the

evaluation metric [2]. MT systems are particularly helpful as they greatly simplify human-to-human communications [3]. The accessibility, quickness, affordability, and user-friendliness of MT systems are its major benefits [4]. Artificial intelligence quickly improves the MT for natural languages like Hindi, English, and Spanish [5]. English and Hindi are the two languages that are the most used worldwide [6]. There are many websites with news articles in both English and Hindi. Latent dirichlet allocation (LDA) conducted numerous studies for undefined text classification [7]. A significant basic tool for NLP is a morphological analyzer and generator (MAG). This is particularly for used for analysing morphologically rich languages like Tamil, where morphemes are utilized to represent different forms of information involving aspect, tense, gender, number, and person [8].

MT performance depends on an availability of a good and large-quality bilingual parallel corpus that is employed for training the system [9]. Natural language is used by the computer to communicate with the user [10]. NLP also deals with handling data across several channels such as television, print, communication, and radio [11]. Multi-modal MT involves analyzing text and visual data before adding it to the translation models to provide high-quality translation outcomes. By including visual data throughout the translation process, the text understands the context information more effectively, establishes ambiguous terms in the phrase, and translates the target message more accurately [12]. The alphabet and numerals of American sign language (ASL) characters are considered separately for the recognition performance [13]. The two most popular MT architectures for language translation are neural machine translation (NMT) and statistical machine translation (SMT). NMT is an end-to-end-neural network-based encoder-decoder model that forecasts the probability of a sequence of words, whereas the SMT uses a log-linear model that consists of translation and language models [14]. In the structure of the encoder-decoder, the encoder compresses the sequences of input in NMT systems into a single vector representation. The decoder then employs the vector representation to create the sequence of outcomes [15]. However, they disregard the significance of semantic data encoded in the text features. Kandimalla *et al.* [16] implemented an English-to-Indian language MT system by employing back translation for Hindi and Bengali to enhance the quality of baseline English-to-Hindi and English-to-Bengali translation systems. The translation results were first assessed employing metrics for automatic evaluation and after then assessed manually to determine the effectiveness of automatic evolution. For both the NMT systems of English-to-Hindi and English-to-Bengali, the back translation technique improved the BLEU scores. However, during the translation of complicated statements or idiomatic phrases, back translation led to the loss of meaning and context. Singh *et al.* [17] presented an MT system for Sanskrit-to-Hindi translation. The method developed a NMT system using linguistic data from a rule-based feed. The method was innovative and suitable to any language of low resource with rich morphology and covered multiple domains, requiring minimum human involvement. The method achieved a high performance by using both human and automatic measures, and also generated effectively in terms of accuracy, response time, and speed. However, MT systems commonly operated on a sentence-by-sentence basis, but struggled to incorporate contextual data effectively.

Garg *et al.* [18] introduced an NMT system to assess out-of-vocabulary (OOV) and multi-word expressions (MWE) in Punjabi-to-English system. To train the various systems of NMT, a parallel corpus for Punjabi-to-English with MWEs was created and utilized. Human evaluation was used to evaluate the NMT for fluency, overall rating, and adequacy. The implemented method improved the overall accuracy of the MT system during Punjabi-to-English translation. However, the NMT system was unable to handle the numerous dialects of Punjabi. Dugonik *et al.* [19] presented a hybrid machine translation (HMT) system that combined NMT and SMT to enhance the NMT's quality. For the Slovenian-English language pair, two NMT and SMT systems were established, each for translation in one direction. The original sentences and translations were placed in the space of the same vector using a multilingual language model. HMT provided a higher-quality translation by utilizing the best characteristics of each system. However, multiple MT techniques were required for establishing and maintaining an HMT system. Eo *et al.* [20] introduced a three-automatic word-level pseudo-quality estimation (QE) construction technique for Korean-English NMT by deploying a parallel or monolingual corpus. To improve the language of pre-trained multilingual models like bidirectional auto-regressive transformer (BART), and XLM-RoBERTa, these distinct pseudo-QE datasets were used. The introduced method was executed to any language pair provided by Google Translate, with an associated corpus for each data-building process. However, the pseudo-QE data construction method was limited to large-scale datasets due to the time, cost and human labor. Tran *et al.* [21] presented an automated collection method to construct a Chinese-Vietnamese bilingual corpus from the website of the Globse multilingual dictionary. To extract the parallel sentence pairs from the website of bilingual, four steps were carried out: obtaining the article material in Chinese and Vietnamese, determining similar Chinese and Vietnamese text pairs, determining identical Chinese Vietnamese sentence pairs from similar text pairs, and post-processing. The presented automated collection method was applied to numerous language pairs apart from Chinese-Vietnamese. However, combining the factors of segmentation and non-segmentation, as well as the benefits

of SMT and NMT was required to enhance the MT translation. Sitender and Bawa [22] implemented a Sanskrit-to-English MT system by employing a hybridized form of rule-based and direct MT method. This method included the language difference between Sanskrit and English, as well as a potential remedy to manage the differences. The Elasticsearch method improved the MT system's ability to obtain information from multiple data dictionaries and rule bases utilized for the system's development. The implemented method achieves a fluency score, BLEU score, and adequacy score using NLP. However, high-quality parallel training data for Sanskrit-to-English translation was inadequate. There were some limitations with the existing methods such as disregarding the significance of semantic data encoded in the text features. To overcome these issues, the Multimodal NMT method is proposed to fully utilize semantic text features on the architecture of NMT, as well as minimize the testing and training time. In the following, the primary contributions of this paper are summarized:

- The multimodal NMT method is proposed for Sanskrit-Hindi translation. The main goal of the method is to fully utilize semantic text features on the NMT architecture and to minimize testing and training time.
- The aforementioned concept is validated on two different NMT architectures: RNN and SAN. The experimental outcomes indicate that the MNMT enhances over baselines on both RNN and SAN architectures.
- Sanskrit to Hindi translations are assessed based on BLEU, and METEOR to evaluate the robustness of the MNMT.

The remainder of the paper is set out as follows: the proposed method is discussed in section 2. Section 3 represents the research method, while section 4 details the results. The conclusion is shown in section 5.

## 2. PROPOSED METHOD

Multimodal neural machine translation (MNMT) is proposed for Sanskrit-Hindi translation. It includes the Corpora dataset, pre-processing, training, testing, MNMT, and evaluation output. The overview of the proposed method is represented in Figure 1.

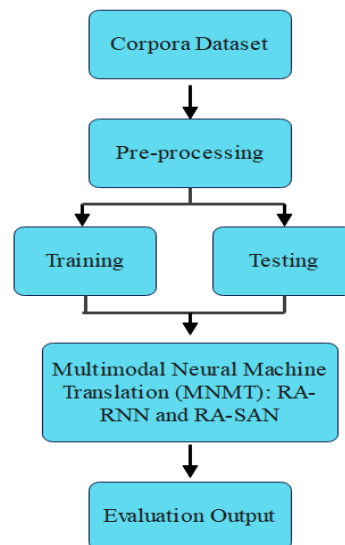


Figure 1. Block diagram for the proposed method

### 2.1. Sanskrit to Hindi Corpora dataset

A Corpora dataset is a kind of structured learning data that includes texts from a variety of sources including Wikipedia, the news, literature, tourism, judicial, healthcare, and the general domain. It has four types: monolingual, parallel corpus, para crawl, and samanantar parallel corpora. Corpora consists of a total of 8.8 billion tokens from news crawls across all 11 languages along with Indian English. The Bhagwad-Geeta consists of 700 slokas, converted to Hindi which is also manually developed. Additionally, the Indian languages corpora initiative (ILCI) project made 50,000 Sanskrit-Hindi corpora. The algorithm is trained using the whole parallel corpus of 162,760 parallel sentences. The remaining non-numeric or non-alphabetical tokens, as well as any punctuation marks and non-printable characters, are then eliminated.

**2.2. Data pre-processing**

The corpus data from the aforementioned dataset are pre-processed using clean text and split text. Text is separated into sentences as an essential process in text cleaning. The non-numeric or non-alphabetical tokens as well as any punctuation marks, and non-printable characters are then eliminated in the pre-processing stage. Unicode characters are converted to ASCII value and all uppercase letters become lowercase. For each pair of imported datasets, these operations are performed on each sentence. The splitting operations are then applied to the cleaned data. Different computation graphs are created because the dataset contains sentence pairs of varying lengths. Then, sentences of a similar length are divided into smaller batches after sorting sentences in a batch according to the length of sentence pairs. The training corpus is shuffled periodically by splitting the corpus into maximum batches and then splitting the corpus again into mini-batches. Applying a gradient for the parameter update completes the processing.

**3. REGION-ATTENTIVE MULTIMODAL RECURRENT NEURAL NETWORK AND REGION-ATTENTIVE MULTIMODAL SELF-ATTENTION NETWORK**

**3.1. Process of region-attentive multimodal recurrent neural network**

After pre-processing the corpus data from the dataset, the region-attentive multimodal recurrent neural network (RA-RNN) is used to extract the text into two components, sentences encoder and decoder as shown in Figure 2. An additional attention mechanism is used for integrating the feature of text. The mechanism of text-attention calculates the text content vector  $c_t$  from the phrase of source  $X = (x_1, x_2, x_3, \dots, x_n)$  to the target sentences  $Y = (y_1, y_2, y_3, \dots, y_g)$ . The decoder employs two attention mechanism RNN with conditional gated recurrent unit ( $cGRU$ )<sup>2</sup> for producing target word  $y_t$  and the current hidden state  $s_t$ . A hidden state  $s_t$  is calculated in  $cGRU$  at time step  $t$  in (1) to (4).

$$\hat{\xi}_t = \sigma(W_\xi E_y[y_{t-1}] + U_\xi S_{t-1}) \tag{1}$$

$$\hat{\gamma}_t = \sigma(W_\gamma E_y[y_{t-1}] + U_\gamma S_{t-1}) \tag{2}$$

$$\dot{S}_t = \tanh(W_Y E_y[y_{t-1}] + \hat{\gamma}_t \odot (U S_{t-1})) \tag{3}$$

$$\hat{S}_t = (1 - \hat{\xi}_t) \odot \dot{S}_t + \hat{\xi}_t \odot S_{t-1} \tag{4}$$

Where  $W_\xi, U_\xi, W_\gamma, U_\gamma, W,$  and  $U$  are trainable parameters and  $E_y$  is the target word vector.

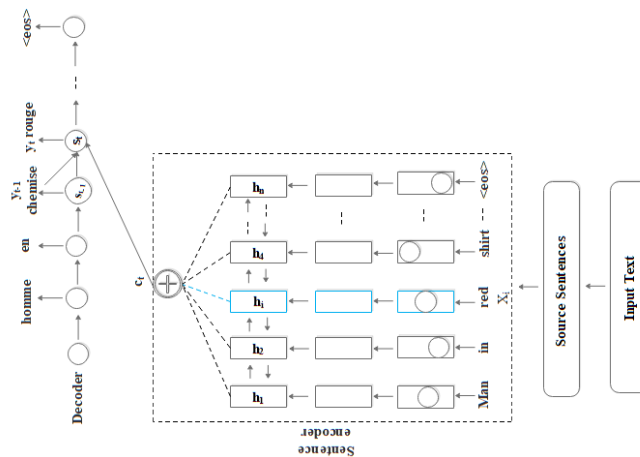


Figure 2. Block diagram for the RA-RNN

**3.1.1. Sentence encoder**

A bidirectional RNN [23] with GRU [24] acts as the sentence encoder. Given a sentence  $X = (x_1, x_2, x_3, \dots, x_n)$ , the forward hidden states with vector annotation  $(\vec{h}_1, \vec{h}_2, \vec{h}_3, \dots, \vec{h}_n)$ , and backward with vector annotation  $(\overleftarrow{h}_1, \overleftarrow{h}_2, \overleftarrow{h}_3, \dots, \overleftarrow{h}_n)$  are updated by encoder. Each  $h_i$  encodes the whole sentences while concentrating on the word  $x_i$  by combining the vectors of forward and backward  $h_i = [\vec{h}_i, \overleftarrow{h}_i]$ , and all words in sentences are referred to as  $C = h_1, h_2, \dots, h_n$ .

**3.1.2. Decoder**

After the encoder, decoder is performed based on the two components, text-attention mechanism and generation. Text-attention mechanism: a  $c_t$  denotes vector of text context vector produced at time step  $t$ . The mathematical formulas are expressed in equations (5) to (7).

$$e_{t,i}^{text} = (V^{text})^T \tanh(U^{text} \hat{s}_t + W^{text} h_i) \tag{5}$$

$$\alpha_{t,i}^{text} = \text{softmax}(e_{t,i}^{text}) \tag{6}$$

$$c_t = \sum_{i=1}^n \alpha_{t,i}^{text} h_i \tag{7}$$

Where  $V^{text}$ ,  $U^{text}$ , and  $W^{text}$  are the trainable parameters,  $e_{t,i}^{text}$  is the attention energy,  $\alpha_{t,i}^{text}$  is the source sentences attention weight matrix.

After the text-attention mechanism, the generation is performed in the second phase. Generation: the new hidden state  $s_t$  is produced in the  $cGRU$  at time step  $t$ . The mathematical formulas are described in (8) to (11).

$$\xi_t = \sigma(W_\xi^{text} c_t + \overline{U}_\xi \hat{s}_t) \tag{8}$$

$$\gamma_t = \sigma(W_\gamma^{text} c_t + \overline{U}_\gamma \hat{s}_t) \tag{9}$$

$$\bar{s}_t = \tanh W^{text} c_t + c_t + \gamma_t \odot (\overline{U}(\hat{s}_t)) \tag{10}$$

$$s_t = (1 - \xi_t) \odot \bar{s}_t + \xi_t \odot \hat{s}_t \tag{11}$$

Where  $W_\xi^{text}$ ,  $\overline{U}_\xi$ ,  $W_\gamma^{text}$ ,  $\overline{U}_\gamma$ ,  $W^{text}$  and  $\overline{U}$  are the parameters of the model,  $\xi_t, \gamma_t$  are respectively the reset/update gate output, and  $\bar{s}_t$  is the updated hidden state.

At last, the probability of the output is calculated as (12).

$$\text{softmax}(L_o \tanh(L_s s_t + L_c c_t + L_z z_t + L_W E_y[y_{t-1}])) \tag{12}$$

Where,  $L_o, L_s, L_c, L_z$ , and  $L_W$  are the trainable parameters.

**3.2. Process of region attentive multimodal self-attention network**

The encoder and decoder are the two components of region attentive multimodal self-attention network (RA-SAN). The architecture of RA-SAN is presented based on the transformers, and the cross-attention mechanism is used throughout the text by the decoder. The block diagram of RA-SAN is represented in Figure 3.

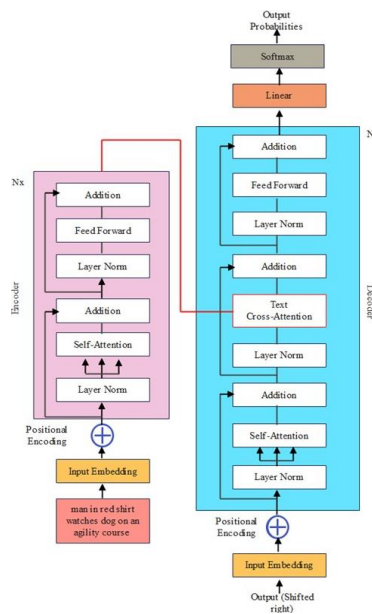


Figure 3. Block diagram for the RA-SAN

### 3.2.1. Encoder

An input layer of embedding serves are used as an index to convert each word in source sentences into a vector representation. Positional encoding is employed for inserting positional data into the input embeddings since the encoder in the transformer lacks recurrence similar to RNN. The N stack identical layers make the encoder where each layer comprises sublayers of feed-forward and self-attention. With the help of the mechanism of multi-head attention, the model simultaneously receives the data from several subspace representations. A vital, fully linked feed-forward network (FFN) is used in the feed-forward sublayer, and is used for each position separately and consistently. The connection of the residual and normalization layers are essential components of the transformer along with two sub-layers. Each of the two sublayers in the model has a residual connection surrounding it, while the residual connection has a normalization layer. Since each sublayer implements a function, the outcome of each sublayer is described as  $(x + \text{sublayer}(\text{LayerNorm}(x)))$ , where  $\text{sublayer}()$  is the function. All embedding layers and sublayers generate dimension  $d_{model}$  outputs to support these residual connections.

### 3.2.2. Decoder

In a stack, the decoder is made up of N identical layers and includes the mechanism of text cross attention that executes multi-head attention on encoder outcome attributes along with two sub-layers. There is a connection between residual over normalization and sublayer like encoder. The summarized outcome of text cross-attention is passed into the FFN sublayer, which is made up of two linear translations with an activation of rectified linear unit (ReLU), as expressed in (13).

$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2 \quad (13)$$

Although the transformations of linear are identical throughout various positions, they employ various parameters from layer to layer, where  $W_1, W_2, b_1,$  and  $b_2$  are trainable parameters. In (13), the input and output dimensions are  $d_{model}$ , and the FFN layer has  $d_{ff}$  dimensions. The decoder finalizes with a layer of linear that performs as a SoftMax layer and classifier to determine the probabilities of the target words.

### 3.3. Text cross attention

As represented in Figures 4 and 5, the mechanism in the transformer of cross-attention serves as a key-value query set mapping to an outcome that is multi-head attention. It executes the function of attention on encoder outcome attributes by utilizing H heads concurrently. Before executing the final linear layer, each head generates an output vector that is concatenated into a single vector.

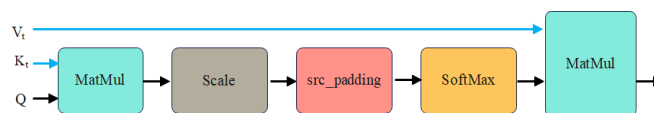


Figure 4. Text scaled dot-product attention

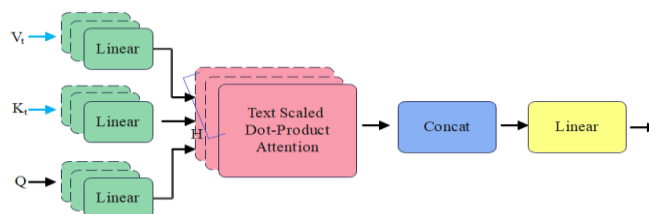


Figure 5. Text multi-head attention

The input consists of queries, dimension key is denoted as  $d_k$ , and dimension value is  $d_v$ . Using dot product multiplication, each query is scaled by  $\sqrt{d_k}$ , multiplied with all keys, and then provided a src-padding to increase the maximum length of the source text. The function of SoftMax is then utilized to determine the value weights. The weighted sum of the values is calculated as the final outcome of scaled dot-product attention. A query's compatibility function with the relevant key is used to determine the weight

provided to each value. The cross-attention is evaluated on a collection of values, keys, and queries at the same time and packed into a matrix  $Q, k_t, V_t$ . The output matrix is calculated in the (14) to (16).

$$Attention(Q, k_t, V_t) = SoftMax\left(\frac{QK_t^T}{\sqrt{dk}}\right)V_t \quad (14)$$

$$MultiHead(Q, k_t, V_t) = Concat(head_t^1, \dots, head_t^H)W^o \quad (15)$$

$$head_t^{i \in [1, H]} = Attention(QW_i^Q, K_tW_i^K, V_tW_i^V) \quad (16)$$

The projections are matrices with the following parameters

$$W_i^Q \in R^{d_{model} \times d_k}$$

$$W_i^K \in R^{d_{model} \times d_k}$$

$$W_i^V \in R^{d_{model} \times d_k}$$

$$W_i^O \in R^{d_{model} \times d_k}$$

## 4. RESULTS

To obtain the efficacy of the multimodal NMT method, this paper employs transformer's settings in toolkit of OpenNMT. In encoder, the number of semantic text features in the RA-RNN and RA-SNN methods is set to  $m=100$ , and  $d_r=2,048$ . The parameters like bilingual evaluation understudy (BLEU) [25] and metric for evaluation of translation with explicit ORdering (METEOR) are utilized for evaluating the model's performance. The numerical expressions for these parameters are expressed in (17) to (19).

### 4.1. Evaluation metrics

- BLEU: it is an essential parameter for assessing sentence translation accuracy in comparison to human-generated reference translation as shown in (17).

$$BLEU = \min\left(1, \frac{output_{length}}{Reference_{length}}\right) \left(\prod_{i=1}^4 precision_i\right) \quad (17)$$

- Metric for evaluation of translation with explicit ORdering (METEOR) - It is a metric to evaluate the output of MT. The metric is based on the harmonic mean of unigram recall and precision as shown in (18) and (19).

$$F_{mean} = \frac{10PR}{9+RP} \quad (18)$$

$$METEOR = F_{mean}(1 - p) \quad (19)$$

### 4.2. Performance analysis

In the RNN-based architecture, N=6 layer is set for the encoder and decoder. The total number of input and output layer's dimensions is set as  $d_{model}=512$ . The value  $d_{ff}=2,048$  is chosen for the inner FFN layer. In both the layers of encoder and decoder, H=8 is set as the head value for each module of multi-head. To decrease dimensions from 2,048 to 512 which are identical dimensions as word embeddings, linear projection on text features is used. On the linear projection, 0.3 dropouts is applied. The smoothing label value is set to 0.1, the sentences-minibatches are set to 40, and the residual and attention dropout is set to 0.3 during training. With a warm-up step of 8,000, the rate of learning is set at 2. The model is trained for 100 epochs before selecting approach with increased BLEU from validation set for accessing test set.

The SAN-based architecture uses a dimension of hidden state 500 for the bi-directional cGRU decoder and GRU encoder, 500 for embedding for source word, 40 for sentences-minibatches, 5 for size of beam, and 0.3 for text dropout. With a learning rate and ADAM of 0.002 for 25 epochs, the model is trained using stochastic gradient descent. Finally, after integration of the validation accuracy and perplexity, the model with the greatest score of BLEU in a validation set is chosen for evaluating test set. Table 1 and Figure 6 show that the RNN-based architecture's experimental results indicate that the RA-RNN achieves both the baselines of text-only RNN and the grid attentive-RNN (GA-RNN) in all translation tasks. This demonstrates that semantic text features improve translation efficiency and utilize better textual information.

Table 2 and Figure 7 exhibit the SAN-based architecture's experimental results. The proposed RA-SAN is compared with their baselines SAN and GA-SAN. RA-SAN accomplishes both the baselines in all tasks of translation. This illustrates that the RA-SAN is universal, resulting in significant performance improvements on various NMT architectures.

Table 1. RNN-based architectures' experimental results

Methods	Sanskrit-to-Hindi	
	BLEU (%)	METEOR (%)
RNN	56.5	71.9
GA-RNN	57.8	72.8
RA-RNN	80.5	75.3

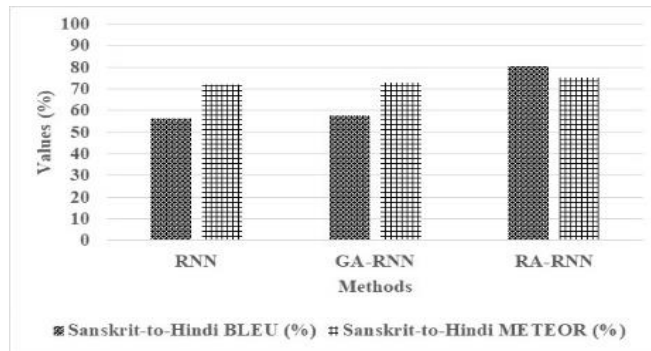


Figure 6. RNN-based architecture experimental results

Table 2. SAN-based architectures experimental results

Methods	Sanskrit-to-Hindi	
	BLEU (%)	METEOR (%)
SAN	57.4	72.2
GA-SAN	59.5	74.4
RA-SAN	78.2	77.1

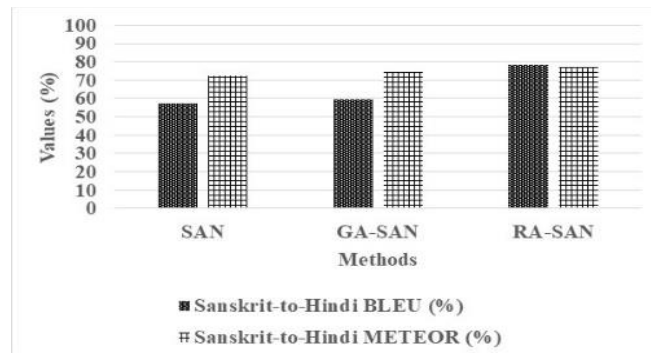


Figure 7. SAN-based architecture experimental results

Table 3 displays the computational cost of several variants that involve the speed of training, decoding, and time of elapse per training epoch. On the translation task of Sanskrit-to-Hindi, all variants with several  $m$  in both methods are tested. The variants' computational cost with  $m=10$ ,  $m=100$  minimum value, and maximum value of  $m=200$  is presented to provide a clear illustration. Overall, as  $m$  increases, the decoding and training speeds decrease concurrently, and the time elapses for each training epoch increases accordingly. Particularly for the same  $m$ , the RA-SAN's decoding and training speeds are fewer than the RA-RNN's. Because the RA-RNN text-attention calculation is additive while the RA-SAN text cross-attention calculation is scaled as a dot-product. So, the RA-SAN computation is higher than RA-RNN.



Table 3. Computational cost of several variants

Methods	RA-RNN			RA-SAN		
	M=10	M=100	M=200	M=10	M=100	M=200
Training	9,140	5,486	3,847	5,575	4,267	2,868
Decoding	7,973	5,296	3,767	5,788	4,549	3,073
Time (s)	52	78	110	72	91	135

### 4.3. Comparative analysis

The comparative analysis includes dataset, methods, BLEU, and METEOR. In this paper, Sanskrit-Hindi corpora dataset is not available, so in the comparative analysis, similar available dataset is compared with the existing methods. Table 4 shows the presented model's comparative evaluation with the existing methods.

The existing method English-to-Indian MT system [16] has two techniques: English-Hindi has a BLEU score of 33.72% and English-to-Bengali has a BLEU score of 11.99%. Sanskrit-Hindi MT system [17] has a BLEU score of 75% and METEOR has 61%. Hybrid MT system [19] has two techniques: English-Slovenian has 42.9%, 61.5% of BLEU and METEOR. Slovenian-English has 47.9 and 70.9 of BLEU and METEOR. When contrasted with the above existing methods, RA-RNN respectively attains a better BLEU and METEOR of 80.5% and 75.3%, and RA-SAN respectively attains a superior BLEU and METEOR of 78.2% and 77.1%, correspondingly.

Table 4. Comparative analysis with existing methods

Author	Dataset	Methods	BLEU (%)	METEOR (%)
Kandimalla <i>et al.</i> [16]	Samanantar parallel corpora	English-to-Indian MT system: English-Hindi	33.72	-
		English-Bengali	11.99	-
Singh <i>et al.</i> [17]	Parallel, monolingual corpora	Sanskrit-Hindi MT system	75	61
Dugonik <i>et al.</i> [19]	Para crawl corpora	Hybrid MT system: English-Slovenian	42.9	61.5
		Slovenian-English	47.9	70.9
Proposed method	Sanskrit-to-Hindi corpora dataset	MNMT: RA-RNN Sanskrit-Hindi	80.5	75.3
		RA-SAN Sanskrit-Hindi	78.2	77.1

### 4.4. Discussion

The advantage of MNMT technique and limitation of existing techniques are discussed in this section. The limitation of the previous technique like English-to-Indian MT system [16] has during the translation of complicated statements or idiomatic phrases, where back translation leads to the loss of meaning and context. Sanskrit-Hindi MT system [17] commonly operates on a sentence-by-sentence basis and struggles to incorporate contextual data effectively. The multiple MT techniques are required for establishing and maintaining HMT [19] system. The presented approach overcomes these existing techniques' limitations. RA-RNN generates an efficient combination of information from various regions in multimodal data which increase the contextual understanding. RA-SAN provides an increased adaptability and flexibility in managing different multimodal inputs, which leads to increase the model performances. By using these methods, the MNMT achieves superior performances. In this paper, the Sanskrit-Hindi corpora dataset is not available, so in the comparative analysis, similar available dataset is compared. The existing methods such as the English-to-Indian MT system, Sanskrit-Hindi MT system, and HMT system are used to justify the efficacy of the MNMT method. When contrasted with the previous methods, the proposed MNMT has two architectures: RA-RNN correspondingly attains 80.5% and 75.3% of BLEU and METEOR, while RA-SAN correspondingly attains 78.2% and 77.1% of BLEU and METEOR. The limitations of the proposed MNMT is high-quality parallel training data for Sanskrit-to-English translation which is inadequate.

## 5. CONCLUSION




In this paper, the MNMT method is proposed for Sanskrit to Hindi translation. The main goal of the method is to fully utilize semantic text features on the NMT architecture, and to minimize testing and training time. The MNMT is validated on two different NMT architectures, RNN and SAN. On the translation task of Sanskrit-to-Hindi, all variants are tested in the methods of both RA-RNN and RA-SAN. The experimental outcomes indicate that MNMT enhances over baselines on both RNN and SAN architecture. The RA-RNN simultaneously achieves a robust BLEU and METEOR values of 80.5% and 75.3%, whereas the RA-SAN simultaneously achieves a robust BLEU and METEOR of 78.2% and 77.1% when compared to the existing methods such as English-to-Indian MT system, Sanskrit-Hindi MT system, and hybrid MT system. In the future, the parts of speech will be identified to develop the correct sentences using deep learning in the MNMT method.

## REFERENCES




- [1] S. A. Mohamed, M. A. Abdou, and A. A. Elsayed, "Residual information flow for neural machine translation," *IEEE Access*, vol. 10, pp. 118313–118320, 2022, doi: 10.1109/ACCESS.2022.3220691.
- [2] K. Mrinalini, V. P., and N. Thangavelu, "SBSim: a sentence-BERT similarity-based evaluation metric for indian language neural machine translation systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1396–1406, 2022, doi: 10.1109/TASLP.2022.3161160.
- [3] M. Brour and A. Benabbou, "ATLASLang NMT: Arabic text language into Arabic sign language neural machine translation," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 9, pp. 1121–1131, Nov. 2021, doi: 10.1016/j.jksuci.2019.07.006.
- [4] N. Koul and S. S. Manvi, "A proposed model for neural machine translation of Sanskrit into English," *International Journal of Information Technology*, vol. 13, no. 1, pp. 375–381, Feb. 2021, doi: 10.1007/s41870-019-00340-8.
- [5] M. A. Islam, M. S. H. Anik, and A. B. M. A. Al Islam, "Towards achieving a delicate blending between rule-based translator and neural machine translator," *Neural Computing and Applications*, vol. 33, no. 18, pp. 12141–12167, Sep. 2021, doi: 10.1007/s00521-021-05895-x.
- [6] R. Singh and S. Singh, "Text similarity measures in news articles by vector space model using NLP," *Journal of The Institution of Engineers (India): Series B*, vol. 102, no. 2, pp. 329–338, Apr. 2021, doi: 10.1007/s40031-020-00501-5.
- [7] A. Srivastav and S. Singh, "Proposed model for context topic identification of English and Hindi news article through LDA approach with NLP technique," *Journal of The Institution of Engineers (India): Series B*, vol. 103, no. 2, pp. 591–597, Apr. 2022, doi: 10.1007/s40031-021-00655-w.
- [8] K. Sarveswaran, G. Dias, and M. Butt, "ThamizhiMorph: a morphological parser for the Tamil language," *Machine Translation*, vol. 35, no. 1, pp. 37–70, Apr. 2021, doi: 10.1007/s10590-021-09261-5.
- [9] S. K. Mahata, A. Garain, D. Das, and S. Bandyopadhyay, "Simplification of English and Bengali sentences for improving quality of machine translation," *Neural Processing Letters*, vol. 54, no. 4, pp. 3115–3139, Aug. 2022, doi: 10.1007/s11063-022-10755-3.
- [10] S. Bhatia, A. Kumar, and M. M. Khan, "Role of genetic algorithm in optimization of Hindi word sense disambiguation," *IEEE Access*, vol. 10, pp. 75693–75707, 2022, doi: 10.1109/ACCESS.2022.3190406.
- [11] D. Draskovic, D. Zecevic, and B. Nikolic, "Development of a multilingual model for machine sentiment analysis in the Serbian language," *Mathematics*, vol. 10, no. 18, p. 3236, Sep. 2022, doi: 10.3390/math10183236.
- [12] X. Shi, Z. Yu, X. Wang, Y. Li, and Y. Niu, "Text-image matching for multi-model machine translation," *The Journal of Supercomputing*, vol. 79, no. 16, pp. 17810–17823, Nov. 2023, doi: 10.1007/s11227-023-05318-9.
- [13] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "CNN based feature extraction and classification for sign language," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 3051–3069, Jan. 2021, doi: 10.1007/s11042-020-09829-y.
- [14] A. Kumar, R. K. Mundotiya, A. Pratap, and A. K. Singh, "TLSPG: transfer learning-based semi-supervised pseudo-corpus generation approach for zero-shot translation," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6552–6563, Oct. 2022, doi: 10.1016/j.jksuci.2022.03.008.
- [15] L. H. Baniata, I. K. E. Ampomah, and S. Park, "A transformer-based neural machine translation model for Arabic dialects that utilizes subword units," *Sensors*, vol. 21, no. 19, p. 6509, Sep. 2021, doi: 10.3390/s21196509.
- [16] A. Kandimalla, P. Lohar, S. K. Maji, and A. Way, "Improving English-to-Indian language neural machine translation systems," *Information*, vol. 13, no. 5, p. 245, May 2022, doi: 10.3390/info13050245.
- [17] 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*, vol. 33, no. 4, pp. 1103–1122, Feb. 2021, doi: 10.1007/s00521-020-04990-9.
- [18] K. D. Garg *et al.*, "Framework for handling rare word problems in neural machine translation system using multi-word expressions," *Applied Sciences*, vol. 12, no. 21, p. 11038, Oct. 2022, doi: 10.3390/app122111038.
- [19] J. Dugonik, M. Sepesy Maučec, D. Verber, and J. Brest, "Reduction of neural machine translation failures by incorporating statistical machine translation," *Mathematics*, vol. 11, no. 11, p. 2484, May 2023, doi: 10.3390/math11112484.
- [20] S. Eo, C. Park, H. Moon, J. Seo, and H. Lim, "Word-level quality estimation for Korean-English neural machine translation," *IEEE Access*, vol. 10, pp. 44964–44973, 2022, doi: 10.1109/ACCESS.2022.3169155.
- [21] P. Tran, T. Nguyen, D.-H. Vu, H.-A. Tran, and B. Vo, "A method of Chinese-Vietnamese bilingual corpus construction for machine translation," *IEEE Access*, vol. 10, pp. 78928–78938, 2022, doi: 10.1109/ACCESS.2022.3186978.
- [22] Sitender and S. Bawa, "A Sanskrit-to-English machine translation using hybridization of direct and rule-based approach," *Neural Computing and Applications*, vol. 33, no. 7, pp. 2819–2838, Apr. 2021, doi: 10.1007/s00521-020-05156-3.
- [23] M. Dhyani and R. Kumar, "An intelligent Chatbot using deep learning with bidirectional RNN and attention model," *Materials Today: Proceedings*, vol. 34, pp. 817–824, 2021, doi: 10.1016/j.matpr.2020.05.450.
- [24] J. Wang, Y. Zhang, L.-C. Yu, and X. Zhang, "Contextual sentiment embeddings via bi-directional GRU language model," *Knowledge-Based Systems*, vol. 235, p. 107663, Jan. 2022, doi: 10.1016/j.knsys.2021.107663.
- [25] L. Benkova, D. Munkova, E. Benko, and M. Munk, "Evaluation of English–Slovak neural and statistical machine translation," *Applied Sciences*, vol. 11, no. 7, p. 2948, Mar. 2021, doi: 10.3390/app11072948.

## BIOGRAPHIES OF AUTHORS






**Prashanth Kammar**    is working as a Assistant Professor of CSE of Prodhadevaraya Institute of Technology, Hosapete. He completed BE and M. Tech (CSE) in UBTD College of Engineering Davanagere. He is doing Research work on natural language processing in Research Centre Smt. Kamala and Sri Venkappa M Agadi College of Engineering and Technology, Lakshmeshwar -582 116, Visvesvaraya Technological University, Belagavi-590018, under the guidance of Dr. Parashuram Baraki, Professor of CS&E and Principal of Smt. Kamala and Sri. Venkappa M Agadi College of Engineering and Technology, Lakshmeshwar and Co-guide Dr. Sunil Kumar G of UVCE, Bengaluru. He can be contacted at email: prashanthkogali@gmail.com.






**Parashuram Baraki**    is working as a Professor of CS&E and Principal of Smt. Kamala and Sri. Venkappa M Agadi College of Engineering and Technology, Lakshmeshwar, Karnataka. He completed Ph.D. in Computer Science and Engineering at Jain University Bangalore Karnataka, under the guidance of Dr. V. Ramaswamy. His research interest includes image processing, video processing and pattern recognition. He can be contacted at email: parashuram.baraki@gmail.com.



**Sunil Kumar Ganganayaka**    has completed Bachelor of Engineering from Visvesvaraya Technological University, Belgaum, Masters of Engineering and Doctoral Degree in Computer Science and Engineering from University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. He has over 15 years of teaching and research experience. Currently he is an Assistant Professor in the Department of CSE, University Visvesvaraya College of Engineering, Bengaluru. He has published over 35 research papers in refereed international journals and conferences. Currently he is guiding 4 Ph.D. students and supervised more than 30 post graduate dissertations in Computer Science and Engineering. He has organized 10 international conferences and more than 20 workshops and FDP's. He has awarded 2 best paper awards and has filed 2 patents. His research interests include cognitive networks, computer networks, wireless sensor networks, parallel and distributed systems, cloud computing and natural language processing. He can be contacted at email: sunil777g@gmail.com.



**Manjunath Swamy Byranahalli Eraiahhas**    graduated from Visvesvaraya Technological University, Belgaum, he obtained his Masters of Engineering and Doctoral Degree from Bangalore University, Bengaluru. Currently he produced 2 Ph.D. students from VTU, Belagavi. He has 15 years of teaching experience. Currently he is working in Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bangalore. He has over 35 research papers to his credit. He received fund from AICTE, VGST and KSCST. He has filed 4 Indian patents and one Australian Patent. His area of interest includes image processing, signal processing, and network security, cloud computing, IoT, data science. He can be contacted at email: manjube2412@gmail.com.



**Kolakaluri Lakshman Arun Kumar**    has completed his Ph.D. in CSE of GITAM University, Visakhapatnam and currently working as a professor in Department of CSE, KNSIT, Bangalore. His research involves in cognitive wireless sensor networks and NLP. He can be contacted at email: drarun.git@gmail.com.