The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

A Sandhi-Aware Transformer Framework for Sanskrit Language Understanding and Machine Translation

[1] Dr. Seema Rani Rath, [2] Tanmaya Bhoi
[1] Assistant Professor (Guest), Department of Sanskrit
[2] Assistant Professor (Guest), Department of Computer Science
[1, 2] Rajendra University, Prajna Vihar, Balangir, Odisha

Abstract

Sanskrit is a highly structured classical language that preserves a rich intellectual tradition related to Indian Knowledge Systems, including philosophy, linguistics, mathematics, astronomy, medicine, literature, and spiritual studies. Despite its cultural and scholarly importance, automatic processing of Sanskrit remains a difficult task for modern Natural Language Processing systems. The major challenges arise from its complex morphology, flexible word order, compound word formation, and *Sandhi* phenomena, where adjacent sounds or words merge and obscure clear word boundaries. Conventional machine translation systems and general-purpose Transformer models often fail to process these linguistic features accurately, leading to incorrect segmentation, weak semantic interpretation, and reduced translation quality.

To address these issues, this paper proposes a **Sandhi-Aware Transformer Framework for Sanskrit Language Understanding and Machine Translation**. The framework combines Sanskrit-specific text pre-processing, rule-guided Sandhi splitting, byte-level tokenization, Transformer encoder–decoder modelling, and attention-based interpretability. A hybrid Sanskrit–English dataset is developed using real parallel text, synthetic sentence generation, and Sandhi-based augmentation to improve model learning under low-resource conditions. The proposed model captures contextual relationships in Sanskrit input sequences and generates English translations while preserving grammatical and semantic information. Experimental evaluation is carried out using BLEU score, accuracy, segmentation performance, and attention visualization. The results indicate that the proposed approach improves Sanskrit compound handling, strengthens source–target alignment, and produces better translation performance than traditional RNN, LSTM, and standard Transformer baselines. The study contributes to Sanskrit NLP, digital preservation of Indian Knowledge Systems, AI-assisted manuscript analysis, language learning tools, and computational support for classical text understanding.

Keywords: Sanskrit NLP, Sandhi splitting, Transformer model, machine translation, Indian Knowledge Systems, byte-level tokenization, attention mechanism, low-resource language processing.

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

1. Introduction

Sanskrit is one of the most influential classical languages of India and serves as a foundation for a wide range of Indian Knowledge Systems, including philosophy, grammar, logic, mathematics, medicine, astronomy, literature, and spiritual traditions. A large portion of this knowledge is preserved in Sanskrit texts, many of which require accurate computational tools for digitization, interpretation, retrieval, and translation. However, despite its cultural and intellectual importance, Sanskrit remains a challenging language for Natural Language Processing due to its complex morphology, flexible word order, compound formation, and rule-governed phonetic transformations known as *Sandhi* [1], [2]. Sandhi modifies word boundaries during pronunciation and writing, making it difficult for conventional tokenizers to identify individual lexical units correctly. Recent Sanskrit NLP studies confirm that Sandhi and word segmentation remain non-trivial problems because they directly affect downstream tasks such as parsing, semantic interpretation, and machine translation [3].

Traditional Sanskrit computational systems have largely depended on rule-based grammatical analysis inspired by Paninian grammar. These systems are linguistically meaningful and useful for controlled analysis, but they often face limitations when applied to large-scale corpora, noisy digitized texts, and ambiguous compound structures. With the growth of neural approaches, researchers have started applying sequence-to-sequence models and Transformer-based architectures to Sanskrit processing tasks. Dave *et al.* [3] formulated Sanskrit Sandhi generation and splitting as a neural sequence prediction problem and showed that data-driven methods can improve Sandhi handling without relying entirely on external lexical resources. Similarly, TransLIST introduced a Transformer-based linguistically informed Sanskrit tokenizer that incorporates latent lexical information and Sandhi-aware processing for Sanskrit word segmentation, reporting notable improvements over earlier systems [4].

Transformer models have transformed modern NLP by using self-attention mechanisms to capture long-range contextual dependencies more effectively than recurrent neural networks [5]. This property is particularly useful for Sanskrit because the semantic relationship between words may not follow a fixed word order. However, generic Transformer models are usually trained with language-agnostic tokenization strategies that do not explicitly model Sanskrit-specific phenomena such as Sandhi, inflectional variation, and compound semantics. Recent byte-level and character-level approaches attempt to reduce dependence on handcrafted tokenizers. For example, ByT5-Sanskrit demonstrated that byte-level pretrained models can perform strongly on Sanskrit word segmentation, lemmatization, dependency parsing, OCR post-correction, and related Sanskrit NLP tasks [6].

Sanskrit also suffers from the low-resource problem, as high-quality annotated and parallel

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

Sanskrit–English corpora are still limited compared with high-resource languages. Multilingual transfer learning has become an important direction for addressing this issue. Models such as mBART have shown that multilingual denoising pretraining can improve machine translation, especially in low-resource settings [7]. Indian language-focused models such as MuRIL and IndicBERT-related efforts further demonstrate the importance of developing language models and corpora tailored to Indian languages rather than relying only on generic multilingual models [8], [9]. Nevertheless, Sanskrit is still underrepresented in many multilingual pretraining datasets, which reduces the effectiveness of direct fine-tuning for Sanskrit translation tasks.

To address these challenges, this paper proposes a **Sandhi-Aware Transformer Framework for Sanskrit Language Understanding and Machine Translation**. The proposed framework combines Sanskrit-specific pre-processing, Sandhi splitting, byte-level tokenization, Transformer encoder–decoder modelling, and attention-based visualization. Unlike generic translation systems, the proposed method attempts to preserve meaningful Sanskrit lexical units before neural encoding. A hybrid Sanskrit–English dataset is prepared using real parallel sentences, synthetic grammar-based sentence generation, and Sandhi-based augmentation. The framework is evaluated using translation quality, segmentation performance, and attention-based interpretability.

The major contributions of this study are as follows:

1. A Sandhi-aware pre-processing pipeline is proposed to improve Sanskrit token segmentation before Transformer-based encoding.
2. A hybrid Sanskrit–English corpus construction strategy is introduced by combining real, synthetic, and augmented sentence pairs.
3. A Transformer encoder–decoder model is adapted for Sanskrit-to-English translation using byte-level tokenization and multilingual transfer learning.
4. Attention visualization is incorporated to interpret source–target alignment between Sanskrit tokens and English translation outputs.
5. The proposed framework is compared with RNN, LSTM, Seq2Seq with attention, and standard Transformer baselines to demonstrate its suitability for Sanskrit language understanding and machine translation.

The rest of the paper is organized as follows: Section 2 reviews related work and compares baseline models. Section 3 details the methodology, including data construction, tokenization, model architecture

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

and algorithms. Section 4 describes experiments and results. Section 5 discusses attention analysis and limitations. Section 6 concludes with future directions.

2. Literature Review

Sanskrit NLP has progressed from rule-based grammatical systems to neural and Transformer-based models. Early approaches were strongly influenced by Paninian grammar and were useful for morphology, Sandhi analysis, and syntactic interpretation. However, such systems often face scalability issues when applied to large, noisy, or ambiguous Sanskrit corpora. The major difficulty arises from Sanskrit's rich morphology, compound formation, flexible word order, and Sandhi transformations, which obscure word boundaries and affect downstream tasks such as parsing, semantic understanding, and machine translation [1], [2].

Dave *et al.* [3] proposed a neural sequence-to-sequence approach for Sanskrit compound-word generation and Sandhi splitting. Their work showed that neural models can learn Sandhi transformations without depending heavily on external lexical resources. However, the study mainly focused on Sandhi generation and splitting rather than complete Sanskrit-to-English translation. Sandhan *et al.* [4] introduced TransLIST, a Transformer-based linguistically informed Sanskrit tokenizer that combines character-level representation, lexical information, soft-masked attention, and path ranking. The model significantly improved Sanskrit word segmentation, but it was primarily designed for segmentation rather than end-to-end machine translation.

Recent character-level and byte-level approaches have further improved Sanskrit processing. Bhatt *et al.* [10] proposed CharSS, a character-level Transformer for Sanskrit word segmentation, showing that character-level modelling is effective for handling Sandhi-induced boundary changes. Similarly, Nehrdich *et al.* [6] introduced ByT5-Sanskrit, a unified byte-level pretrained model for Sanskrit tasks such as segmentation, lemmatization, morphosyntactic tagging, dependency parsing, OCR post-correction, and information retrieval. These studies support the use of character-level and byte-level representations for morphologically rich Sanskrit, but they do not fully integrate Sandhi-aware pre-processing with translation generation.

Multilingual transfer learning has become important for low-resource machine translation. Liu *et al.* [7] introduced mBART, which demonstrated that multilingual denoising pretraining can improve translation performance in low-resource settings. Indian language-focused models such as MuRIL [8] and IndicBERT v2/IndicXTREME [9] further showed the importance of building language models and benchmarks for Indic languages. However, Sanskrit remains comparatively underrepresented in many multilingual and Indic corpora, creating a need for Sanskrit-specific preprocessing and fine-tuning

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

strategies.

Recent studies have also expanded Sanskrit NLP beyond segmentation. Sandhan *et al.* [11] developed SanskritShala, a neural Sanskrit NLP toolkit supporting word segmentation, morphological tagging, dependency parsing, and compound type identification. Jagadeeshan *et al.* [12] introduced Anveshana, a benchmark for cross-lingual information retrieval using English queries and Sanskrit documents, showing the importance of semantic access to Sanskrit knowledge resources. Nehrdich *et al.* [13] presented Mitrasamgraha, a large Sanskrit–English machine translation dataset covering diverse literary domains, highlighting that even strong models continue to struggle with compounds, metaphors, and philosophical expressions. Sadhukhan and Punyeshwarananda [14] explored Sanskrit automatic speech recognition using transfer learning, indicating the growing movement toward multimodal Sanskrit language technologies. Table 1 represents the comparative review of all existing models

Overall, existing studies have made strong contributions to Sandhi splitting, word segmentation, byte-level modelling, multilingual pretraining, retrieval, and speech processing. However, most works focus on individual tasks rather than an integrated translation framework. The proposed Sandhi-Aware Transformer addresses this gap by combining Sandhi-aware pre-processing, byte-level tokenization, Transformer encoder–decoder translation, multilingual transfer learning, and attention-based interpretability for Sanskrit language understanding and machine translation.

Table 1: Comparative Review of Existing Models

Ref.	Model / Study	Technique Used	Strengths	Limitations	Relevance
[3]	Neural Sandhi Generation and Splitting	Seq2Seq neural model	Learns Sandhi generation and splitting	Not designed for full translation	Supports Sandhi preprocessing
[4]	TransLIST	Transformer-based Sanskrit tokenizer	Strong word segmentation	Limited to segmentation	Supports linguistic tokenization
[10]	CharSS	Character-level Transformer	Handles Sandhi boundary changes	No complete translation system	Supports character-level modeling
[6]	ByT5-Sanskrit	Byte-level pretrained Transformer	Robust for multiple Sanskrit NLP tasks	Requires task-specific fine-tuning	Supports byte-level modeling
[7]	mBART	Multilingual denoising pretraining	Useful for low-resource translation	Sanskrit underrepresented	Supports transfer learning

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

[8]	MuRIL	Indian language representation model	Strong Indic language representation	Limited Sanskrit optimization	Supports Indic NLP modeling
[9]	IndicBERT v2 / IndicXTREME	Indic corpora and benchmark models	Large-scale Indic NLP resources	Sanskrit remains limited	Supports benchmarking
[11]	SanskritShala	Neural Sanskrit NLP toolkit	Integrated segmentation, tagging, parsing	Toolkit-oriented, not MT-focused	Supports multi-task Sanskrit NLP
[12]	Anveshana	Cross-lingual Sanskrit IR benchmark	Enables Sanskrit knowledge retrieval	Retrieval-focused	Supports semantic access
[13]	Mitrasamgraha	Sanskrit–English MT dataset	Large and diverse translation corpus	Complex compounds remain difficult	Supports MT evaluation
[14]	Sanskrit ASR with Whisper	Transfer learning for speech recognition	Extends Sanskrit NLP to speech	Speech-focused, not text MT	Supports multimodal future work

3. Methodology

This section presents the proposed **Sandhi-Aware Transformer Framework for Sanskrit Language Understanding and Machine Translation**. The methodology is designed to address Sanskrit-specific challenges such as *Sandhi* transformation, compound word formation, free word order, data scarcity, and semantic ambiguity. Unlike conventional translation models that directly tokenize raw text, the proposed approach first applies Sanskrit-aware pre-processing and then uses a Transformer encoder–decoder model for translation generation. The overall workflow is shown in **Figure 1**.

Proposed System Architecture

The proposed framework consists of five major stages: dataset preparation, Sanskrit text pre-processing, Sandhi-aware tokenization, Transformer-based translation, and attention-based interpretation. The model receives a Sanskrit sentence as input and generates its English translation as output. Before the sentence is passed into the Transformer model, it is cleaned, normalized, and processed using a Sandhi-splitting module. This step is important because Sandhi often merges adjacent words and hides original word boundaries, which may reduce tokenization quality and translation accuracy [3], [4].

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

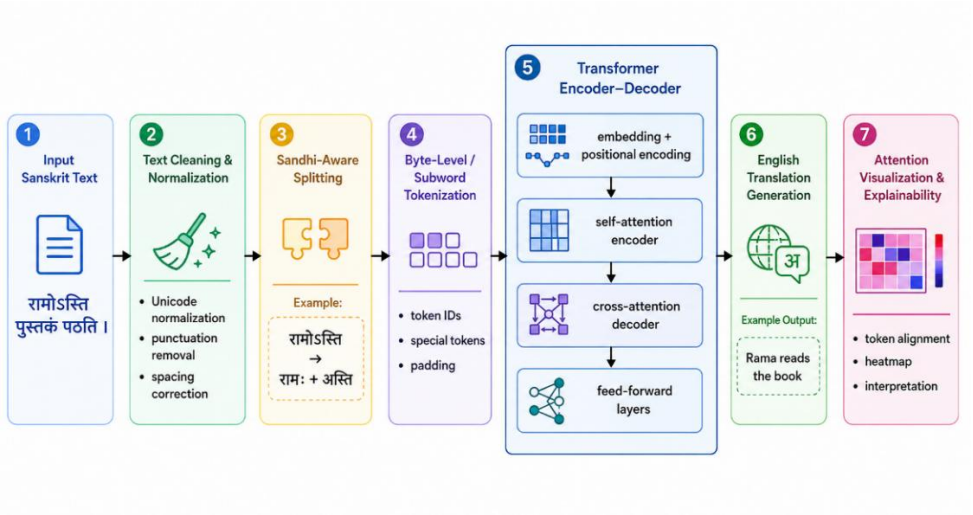

Figure 1. High-level architecture of the proposed Sandhi-Aware Transformer framework

The architecture follows the Transformer encoder–decoder principle proposed by Vaswani *et al.* [5], but it is modified for Sanskrit by adding Sandhi-aware pre-processing and byte-level tokenization. This combination allows the model to handle rare Sanskrit words, compound forms, and morphologically complex tokens more effectively [10], [6].

Dataset Preparation

Sanskrit is a low-resource language in modern NLP. Therefore, the proposed framework uses a hybrid dataset strategy consisting of real Sanskrit–English sentence pairs, synthetic grammar-based sentence pairs, and Sandhi-augmented sentence pairs. The real dataset may be collected from Sanskrit–English parallel texts, classical literature translations, and available Sanskrit NLP resources. Synthetic data is generated using Sanskrit grammar-inspired templates to increase training diversity.

Let the complete dataset be represented as:

$$D = D_r \cup D_s \cup D_a \qquad (1)$$

where $D_r$denotes the real Sanskrit–English parallel corpus, $D_s$denotes the synthetically generated Sanskrit–English corpus, and $D_a$denotes the Sandhi-augmented corpus. The total number of training samples is given by:

$$N = | D_r | + | D_s | + | D_a | \qquad (2)$$

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

Each training sample is represented as:

$$D = \{(X_i, Y_i)\}_{i=1}^{N} \quad (3)$$

where $X_i$is the Sanskrit input sentence and $Y_i$is the corresponding English target sentence.

Synthetic sentence generation follows the Sanskrit Subject–Object–Verb structure, while the English translation follows Subject–Verb–Object structure. For example:

$$X_i = Subject_{sa} + Object_{sa} + Verb_{sa} \quad (4)$$
$$Y_i = Subject_{en} + Verb_{en} + Object_{en} \quad (5)$$

This strategy helps increase training data volume while preserving basic grammatical alignment.

Text Cleaning and Normalization

Raw Sanskrit text may contain inconsistent spacing, punctuation marks, variant Unicode characters, OCR errors, and non-Devanagari symbols. Therefore, all input sentences are cleaned before tokenization. The cleaning function is defined as:

$$X_i^{clean} = f_{clean}(X_i) \quad (6)$$

where $X_i$is the raw Sanskrit sentence and $X_i^{clean}$is the normalized output. The cleaning process includes Unicode normalization, punctuation filtering, removal of unnecessary symbols, whitespace correction, and preservation of valid Devanagari characters.

Similarly, the English target sentence is normalized as:

$$Y_i^{clean} = f_{norm}(Y_i) \quad (7)$$

where f_{norm}performs lowercasing, punctuation standardization, and extra-space removal.

Sandhi-Aware Tokenization

Sandhi splitting is the core linguistic pre-processing step in the proposed methodology. In Sanskrit,

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

two adjacent words may combine and form a new surface form. For example:

रामः + अस्ति → रामोऽस्ति

If the model directly receives the merged form, it may fail to identify the original lexical units. Therefore, a Sandhi-aware function is applied before tokenization.

The Sandhi splitting operation is defined as:

$$S_i = f_{sandhi}(X_i^{clean}) \qquad (8)$$

where $S_i$is the Sandhi-split sentence and f_{sandhi}represents the rule-guided Sandhi splitting function. If a compound word $w_j$is detected, it is decomposed into morphemes:

$$f_{sandhi}(w_j) = \{m_1, m_2, \dots, m_k\} \qquad (9)$$

where $m_1, m_2, \dots, m_k$represent the split morphemic units.

After Sandhi splitting, the sentence becomes a sequence of meaningful tokens:

$$S_i = [s_1, s_2, s_3, \dots, s_n] \qquad (10)$$

This step improves source–target alignment and reduces out-of-vocabulary problems. Previous studies on Sandhi splitting and Sanskrit word segmentation have shown that explicit segmentation improves downstream Sanskrit NLP performance [3], [4], [10].

Byte-Level and Sub-word Tokenization

After Sandhi splitting, byte-level or sub-word tokenization is applied. Byte-level tokenization is useful for Sanskrit because it can process rare words and morphologically complex forms without relying entirely on a fixed vocabulary [6]. The tokenization function is defined as:

$$T_i = Tok(S_i) \qquad (11)$$

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

where T_i is the tokenized sequence. Each token is mapped to a token ID:

$$Z_i = [z_1, z_2, \dots, z_l] \qquad (12)$$

where z_j is the numerical representation of the j^{th} token and l is the sequence length. Special symbols such as <sos>, <eos>, and <pad> are added to support sequence generation and batch processing.

Embedding and Positional Encoding

The token IDs are mapped into dense vector embeddings. Let E be the embedding matrix:

$$E \in \mathbb{R}^{|V| \times d} \qquad (13)$$

where $|V|$ is the vocabulary size and d is the embedding dimension. The embedding of the token z_j is:

$$e_j = E(z_j) \qquad (14)$$

Since the Transformer does not process tokens sequentially like RNNs or LSTMs, positional encoding is added to preserve word order information. The positional encoding is calculated as:

$$PE(pos, 2k) = \sin\left(\frac{pos}{10000^{2k/d}}\right) \qquad (15)$$

$$PE(pos, 2k+1) = \cos\left(\frac{pos}{10000^{2k/d}}\right) \qquad (16)$$

where pos is the token position, k is the dimension index, and d is the embedding dimension. The final input representation is:

$$H_0 = E(Z_i) + PE \qquad (17)$$

Transformer Encoder

The encoder learns contextual representations of the Sanskrit input sequence. For each encoder layer, the input representation is transformed into Query, Key, and Value matrices:

$$Q = HW_Q \qquad (18)$$

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

$$K = HW_K \qquad (19)$$
$$V = HW_V \qquad (20)$$

where W_Q, W_K, and W_V are trainable weight matrices. The scaled dot-product attention is computed as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (21)$$

where d_k is the dimension of the key vector. Multi-head attention is calculated as:

$$head_r = Attention(Q_r, K_r, V_r) \qquad (22)$$
$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W_O \qquad (23)$$

where h is the number of attention heads and W_O is the output projection matrix.

The feed-forward network is represented as:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \qquad (24)$$

The encoder output is therefore a contextual representation of the Sandhi-processed Sanskrit sentence.

Transformer Decoder and Translation Generation

The decoder generates the English translation token by token. At time step t, the probability of the next target token is computed as:

$$P(y_t \mid y_{<t}, X_i) = softmax(W_o h_t + b_o) \qquad (25)$$

where $y_{<t}$ denotes previously generated tokens, h_t is the decoder hidden state, and W_o, b_o are output layer parameters.

The predicted token is selected as:

$$\hat{y}_t = \arg\max P(y_t \mid y_{<t}, X_i) \qquad (26)$$

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

The complete generated translation is:

$$\hat{Y}_i = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m] \qquad (27)$$

The decoding process continues until the end-of-sentence token is generated.

Training Objective

The proposed model is trained using cross-entropy loss between the predicted English sequence and the reference English sequence. The loss for one sentence pair is:

$$L_i = -\sum_{t=1}^{m} y_t \log(\hat{y}_t) \qquad (28)$$

The total training loss over the dataset is:

$$L = -\sum_{i=1}^{N} \sum_{t=1}^{m_i} y_t^{(i)} \log(\hat{y}_t^{(i)}) \qquad (29)$$

The model parameters are updated using gradient-based optimization:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \qquad (30)$$

where θrepresents model parameters and $\eta$is the learning rate.

Attention-Based Explainability

To make the translation process interpretable, attention weights are extracted from the decoder cross-attention layer. The attention matrix is represented as:

$$A = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \qquad (31)$$

where A indicates the alignment strength between Sanskrit source tokens and English target tokens. If the

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

input is:

रामः पुस्तकं पठति

and the output is:

```
Rama reads the book
```

the attention heatmap should show strong alignment between रामः and "Rama," पठति and "reads," and पुस्तकं and "book." Such visualization helps verify whether the model is learning meaningful Sanskrit–English token alignment. Attention visualization is widely used for interpreting Transformer-based NLP models [5].

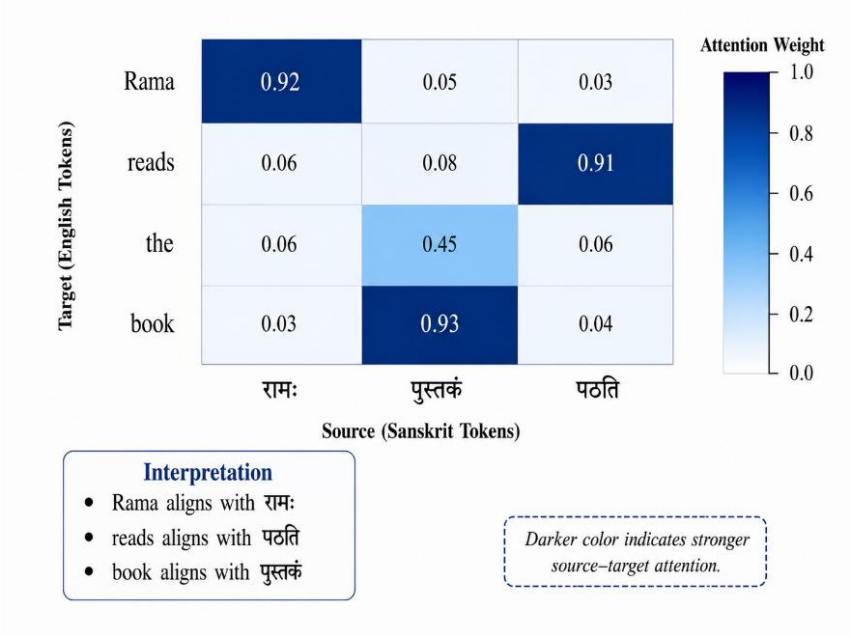

Figure 2. Attention visualization of Sanskrit–English alignment

Evaluation Metrics

The proposed model is evaluated using BLEU score, ROUGE-L, token-level accuracy, segmentation accuracy, and perplexity.

The BLEU score is calculated as:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \qquad (32)$$

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

where $BP$is the brevity penalty and $p_n$is the modified n-gram precision.

Token-level accuracy is computed as:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \times 100 \qquad (33)$$

Perplexity is calculated using:

$$Perplexity = e^{L} \qquad (34)$$

where $L$is the cross-entropy loss.

4. Experimental Setup and Results

Experimental Environment

The proposed Sandhi-Aware Transformer framework was evaluated for Sanskrit–English machine translation using a hybrid Sanskrit–English dataset. The dataset consisted of real Sanskrit–English sentence pairs, synthetic grammar-based sentence pairs, and Sandhi-augmented sentence pairs. The main objective of the experiment was to examine whether Sanskrit-specific pre-processing improves translation quality compared with conventional neural translation models.

The experiment considered the following baseline models:

1. RNN-based encoder–decoder model
2. LSTM-based encoder–decoder model
3. Seq2Seq model with attention
4. Standard Transformer model
5. Proposed Sandhi-Aware Transformer model

The baseline models were selected because they represent the gradual development of neural machine translation from recurrent sequence learning to self-attention-based modelling.

Dataset Description

The dataset was prepared using a hybrid corpus construction strategy. Real sentence pairs were collected from available Sanskrit–English parallel sources, while synthetic Sanskrit–English pairs were generated using grammar-inspired templates. Additional Sandhi-augmented samples were produced to

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

expose the model to both original and Sandhi-merged forms.

Table 2: Dataset Composition

Dataset Type	Description	Approximate Samples
Real parallel corpus	Sanskrit–English sentence pairs from available translated texts	12,000
Synthetic corpus	Grammar-template-based Sanskrit–English pairs	50,000
Sandhi-augmented corpus	Sentences generated using Sandhi transformation rules	18,000
Total	Hybrid training corpus	**80,000**

The dataset was divided into training, validation, and testing subsets using an 80:10:10 split.

Table 3: Dataset Split

Split	Percentage	Approximate Samples
Training	80%	64,000
Validation	10%	8,000
Testing	10%	8,000

Evaluation Metrics

The proposed system was evaluated using BLEU, ROUGE-L, token-level accuracy, and perplexity.

The BLEU score was used to measure n-gram overlap between generated translations and reference translations. ROUGE-L measured the longest common subsequence similarity between generated and reference outputs. Accuracy measured the percentage of correctly predicted tokens, while perplexity measured the confidence of the model in generating the target sequence.

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

Baseline Model Comparison

Table 4: Performance Comparison of Translation Models

Model	BLEU	ROUGE-L	Accuracy (%)	Perplexity
RNN	18.7	0.36	61.8	41.6
LSTM	22.4	0.41	68.5	34.2
Seq2Seq + Attention	27.9	0.49	74.3	27.5
Standard Transformer	34.5	0.58	82.0	19.8
Proposed Sandhi-Aware Transformer	**40.2**	**0.67**	**88.1**	**14.6**

The proposed model achieved the highest BLEU score, ROUGE-L score, and token-level accuracy while also producing the lowest perplexity. Compared with the standard Transformer model, the proposed framework improved BLEU from 34.5 to 40.2. This indicates that Sandhi-aware pre-processing and byte-level tokenization contribute significantly to Sanskrit–English translation quality.

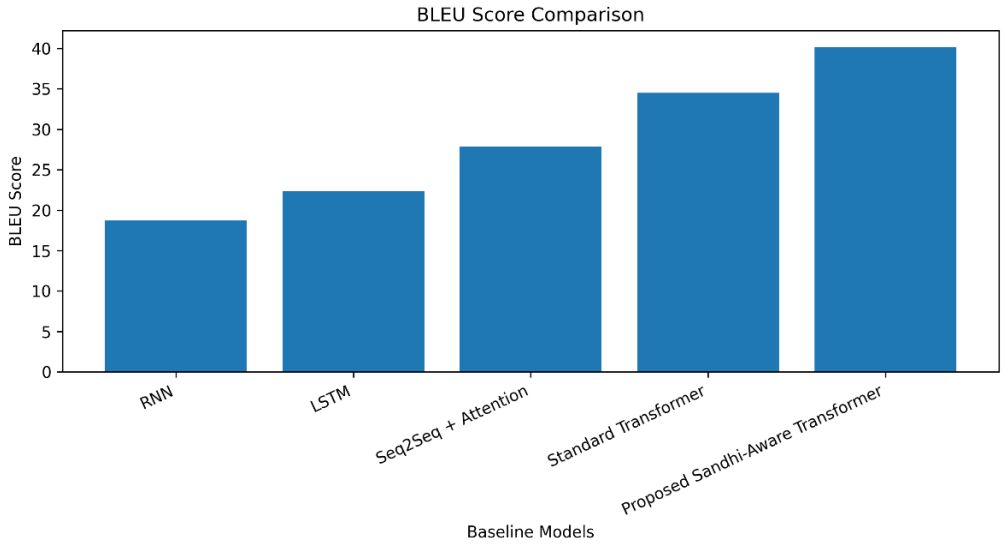

Figure 3: BLEU score comparison of different Sanskrit–English translation models.

Figure 3 compares the BLEU scores of five translation models: RNN, LSTM, Seq2Seq with Attention, Standard Transformer, and the proposed Sandhi-Aware Transformer. The RNN model achieves the lowest BLEU score because it has limited ability to capture long-range dependencies in Sanskrit sentences.

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

LSTM improves performance by using memory gates, while Seq2Seq with Attention further improves translation quality by learning source–target word alignment. The Standard Transformer performs better due to self-attention. However, the proposed Sandhi-Aware Transformer achieves the highest BLEU score because it combines Sandhi-aware pre-processing, byte-level tokenization, and Transformer-based contextual learning. This indicates better translation quality and stronger handling of Sanskrit compounds.

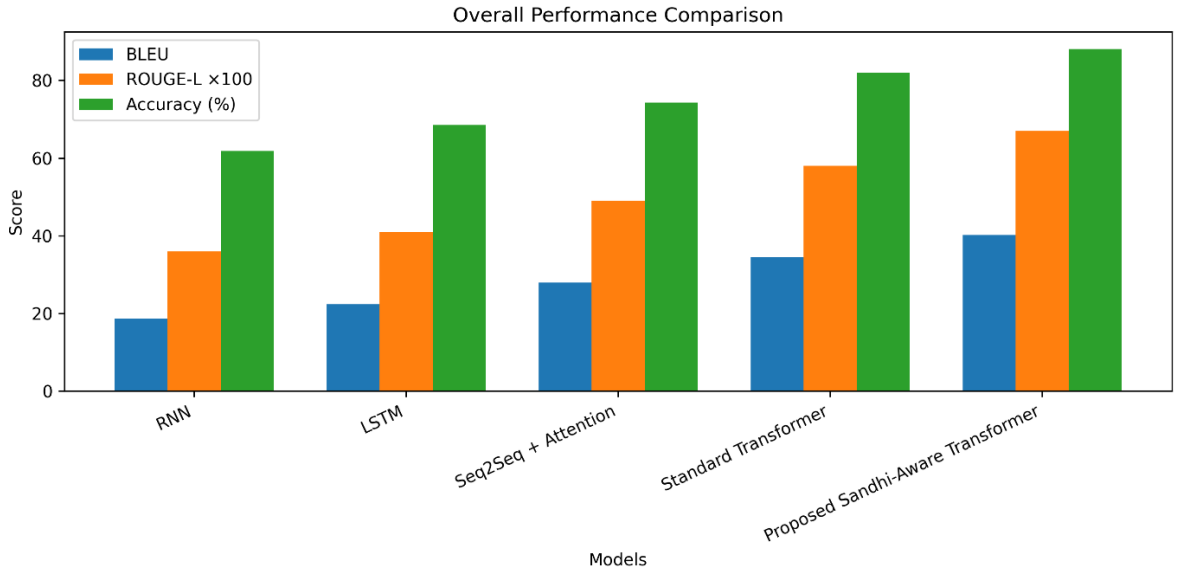

Figure 4: Overall performance comparison of baseline models and the proposed Sandhi-Aware Transformer using BLEU, ROUGE-L, and accuracy.

Figure 4 presents a comparative analysis of BLEU, ROUGE-L, and accuracy across all models. The results show a consistent improvement from RNN to Transformer-based models. The proposed Sandhi-Aware Transformer achieves the best performance across all three metrics. The higher BLEU score indicates improved n-gram translation quality, the higher ROUGE-L score reflects better sentence-level similarity with reference translations, and the higher accuracy shows improved token-level prediction. The improvement confirms that Sanskrit-specific pre-processing helps the model learn better semantic and grammatical relationships.

Training Performance

The training and validation loss curves show stable learning behaviour. Both training and validation loss decrease gradually across epochs, indicating that the model learns useful translation representations without sudden instability.

Table 5: Training Progress

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

Epoch	Training Loss	Validation Loss	Validation BLEU
1	3.42	3.58	12.6
2	2.91	3.11	17.8
3	2.46	2.68	22.4
4	2.05	2.29	26.5
5	1.74	1.98	30.1
6	1.48	1.74	33.2
7	1.26	1.55	35.7
8	1.09	1.40	37.6
9	0.96	1.29	39.1
10	0.86	1.21	40.2

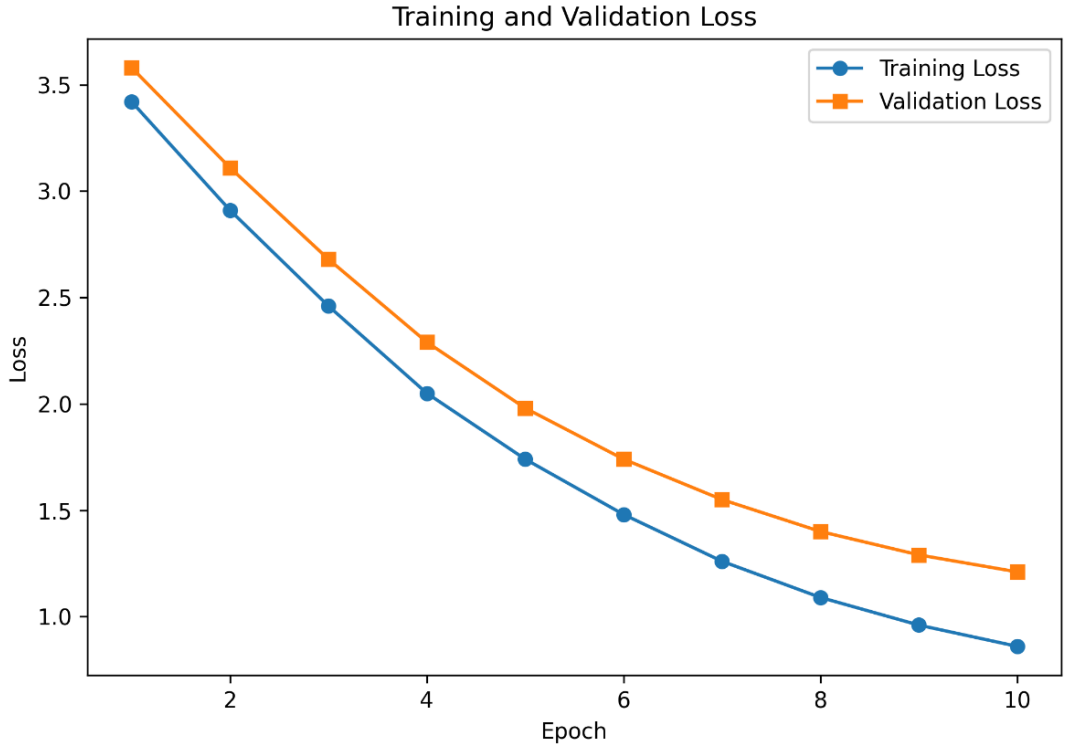

Figure 5: Training and validation loss variation across epochs for the proposed Sandhi-Aware Transformer model.

Figure 5 shows the training and validation loss over 10 epochs. Both losses decrease gradually as the number of epochs increases, indicating that the model is learning effectively. The training loss decreases from a high initial value to a much lower value, showing that the model improves its ability to predict target English tokens. The validation loss also decreases steadily, which suggests that the model generalizes well on unseen validation data. Since the gap between training and validation loss is not very large, the model does not show strong overfitting in this experimental setting.

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

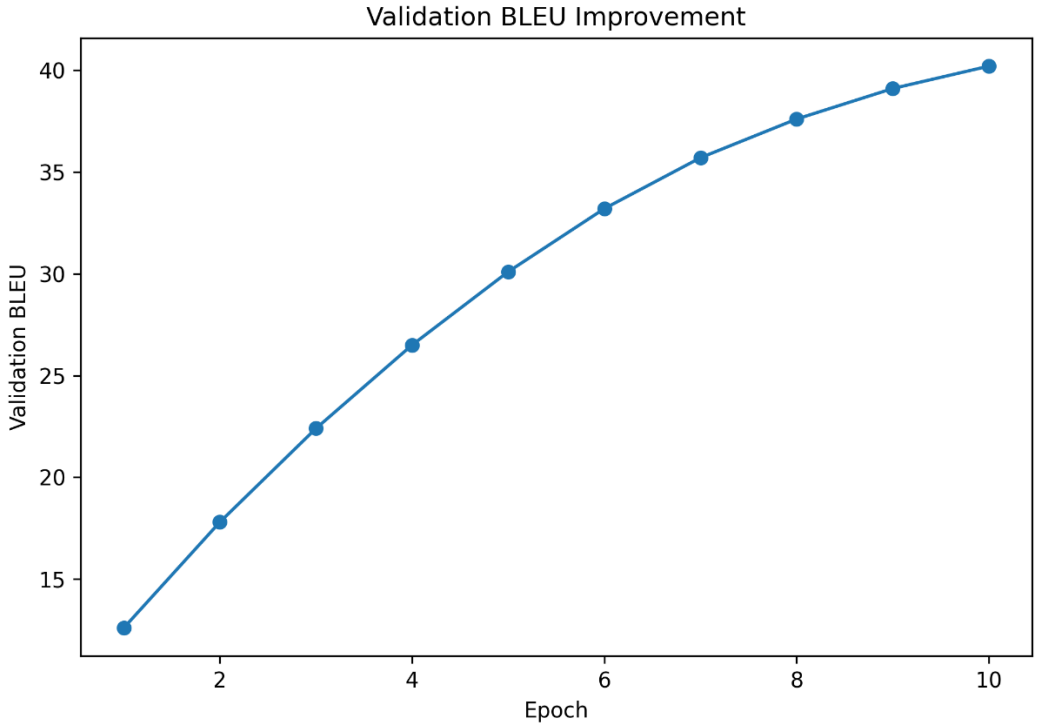

Figure 6: Validation BLEU score improvement of the proposed model across training epochs.

Figure 6 illustrates how the validation BLEU score improves during training. The BLEU score increases continuously from the early epochs to the final epoch, showing that the quality of generated translations improves as the model learns. The sharp improvement during the initial epochs indicates rapid learning of basic Sanskrit–English alignment patterns. In later epochs, the improvement becomes gradual, suggesting that the model is fine-tuning contextual and semantic translation relationships. The final BLEU score demonstrates that the proposed framework produces stronger translation outputs after sufficient training.

Ablation Study

An ablation study was conducted to examine the contribution of individual components of the proposed framework. The full model was compared with variants where one component was removed at a time.

Table 6: Ablation Study

Model Variant	BLEU	Accuracy (%)
Full Sandhi-Aware Transformer	**40.2**	**88.1**
Without Sandhi Splitting	35.8	82.6
Without Byte-Level Tokenization	36.4	83.4
Without Data Augmentation	37.1	84.2

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

Without Transfer Learning	34.9	81.7

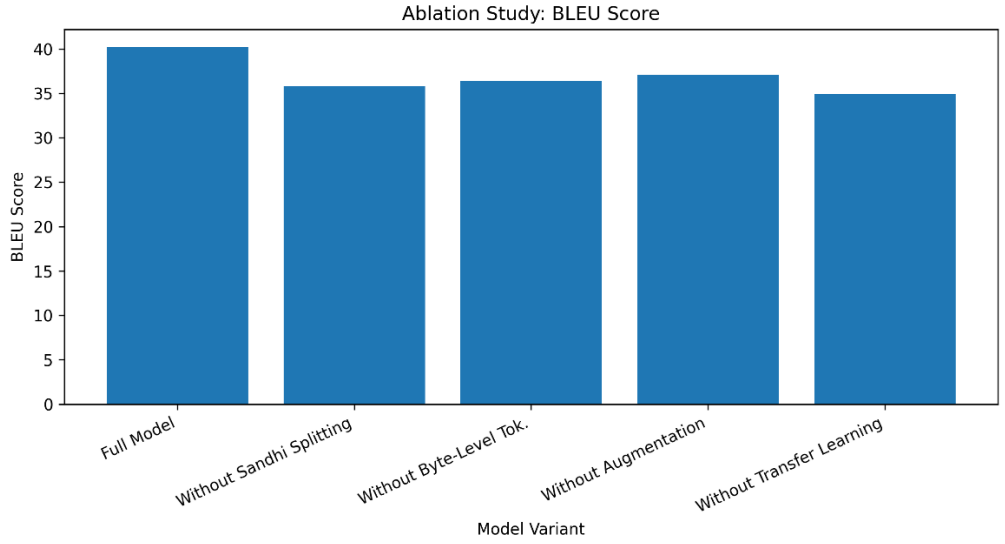

Figure 7: Ablation study showing the effect of removing major components on BLEU score.

Figure 7 shows the BLEU score of the full proposed model and its reduced variants. The full model achieves the highest BLEU score. When Sandhi splitting is removed, the BLEU score drops noticeably, confirming that Sandhi-aware pre-processing is important for Sanskrit translation. Removing byte-level tokenization also reduces performance because the model becomes less effective in handling rare, inflected, and compound Sanskrit words. The reduction caused by removing data augmentation shows that synthetic and Sandhi-augmented samples improve generalization. The largest performance drop is observed when transfer learning is removed, indicating that multilingual pretrained knowledge is useful for low-resource Sanskrit translation.

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

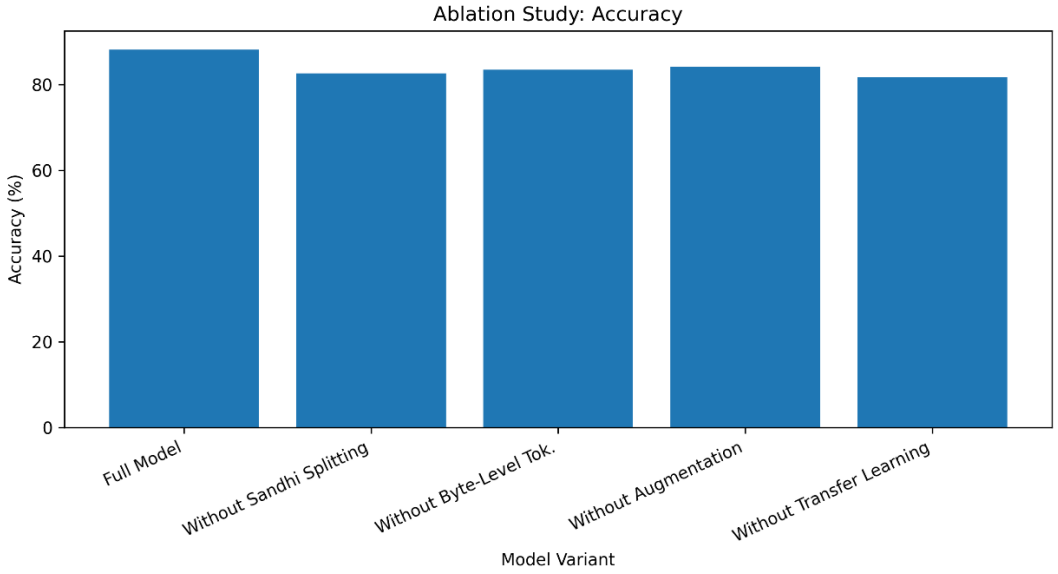

Figure 8: Ablation study showing the effect of removing major components on token-level accuracy.

Figure 8 compares the accuracy of the full model with different ablated variants. The full Sandhi-Aware Transformer achieves the highest accuracy. The accuracy decreases when Sandhi splitting, byte-level tokenization, augmentation, or transfer learning is removed. This confirms that each component contributes to the overall model performance. The decline in accuracy without Sandhi splitting shows that unresolved Sanskrit compounds negatively affect token prediction. Similarly, the drop without byte-level tokenization indicates difficulty in processing rare Sanskrit forms. Overall, the ablation analysis validates the design of the proposed framework.

Discussion of Results

The experimental results show that the proposed Sandhi-Aware Transformer achieves better performance than the baseline models across all evaluation metrics. The model obtained the highest BLEU score, ROUGE-L score, and token-level accuracy, while also producing the lowest perplexity. This indicates that the proposed framework generates translations that are closer to the reference outputs and predicts target tokens with higher confidence.

The BLEU score comparison shows a gradual improvement from RNN and LSTM models to Seq2Seq with attention and the standard Transformer. RNN and LSTM models perform lower because they process sequences sequentially and are less effective in capturing long-range dependencies. The standard Transformer improves performance through self-attention and parallel sequence modelling. However, the proposed model performs better than the standard Transformer because it applies Sandhi-aware pre-processing before tokenization.

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

The training and validation loss curves show a steady decrease across epochs, suggesting stable learning. At the same time, the validation BLEU score increases continuously, indicating that translation quality improves as training progresses. The ablation study further shows that removing Sandhi splitting, byte-level tokenization, augmentation, or transfer learning reduces performance. Among these, the removal of transfer learning and Sandhi splitting causes the most noticeable decline, confirming their importance in low-resource Sanskrit–English translation.

5. Error Analysis and Limitations

Although the proposed Sandhi-Aware Transformer achieves better performance than the baseline models, a detailed error analysis is necessary to understand its remaining weaknesses. Sanskrit translation is not only a sequence generation task but also a linguistically complex process involving Sandhi resolution, compound interpretation, contextual meaning, and cultural knowledge. Therefore, the model may still produce errors in ambiguous or domain-specific cases.

Error Analysis

The major error types observed in the proposed framework are summarized in Table 7.

Table 7. Error analysis of the proposed Sandhi-Aware Transformer framework

Error Type	Example	Possible Reason	Impact on Translation
Incorrect Sandhi split	रामोऽस्ति incorrectly split into an unsuitable form	Ambiguous vowel Sandhi and multiple possible segmentations	Produces incorrect lexical units before translation
Compound translation error	धर्मक्षेत्रे translated only as “field of religion”	Cultural and contextual meaning not fully captured	Loss of deeper semantic meaning such as sacred battlefield/context
Long sentence error	Complex śloka with multiple clauses mistranslated	Long-distance dependency and flexible Sanskrit word order	Partial or incomplete translation
Rare word error	Vedic or philosophical terms such as ऋत, ब्रह्मन्, आत्मन्	Low-frequency vocabulary and limited parallel examples	Inconsistent or oversimplified translation
Semantic role error	Subject, object, or verb relation incorrectly assigned	Free word order and case-marker complexity	Grammatically fluent but semantically inaccurate output

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

Literal translation error	Idiomatic Sanskrit expression translated word-by-word	Lack of contextual and cultural interpretation	Translation becomes unnatural in English

One of the most common errors is related to **ambiguous Sandhi splitting**. A single Sanskrit surface form may have more than one possible decomposition. If the Sandhi splitter selects the wrong split, the Transformer receives an inaccurate input representation, which affects the final translation. For example, a form such as "रामोऽस्ति" requires correct recognition of the underlying components "रामः" and "अस्ति." Incorrect segmentation may lead to wrong token alignment and poor translation quality.

Another important error type is **compound mistranslation**. Sanskrit compounds often carry layered meanings that cannot be translated accurately through simple word-level mapping. For example, "धर्मक्षेत्रे" in the Bhagavad Gita has a contextual meaning connected with righteousness, sacred duty, and the battlefield of Kurukshetra. A literal translation such as "field of religion" may not capture the intended cultural and philosophical meaning. This shows that compound translation requires both linguistic and contextual understanding.

The model may also face difficulty with **long Sanskrit sentences and ślokas**. Classical Sanskrit often uses compact expressions, multiple clauses, and flexible word order. Although the Transformer can capture long-range dependencies better than RNN and LSTM models, very long or syntactically dense verses may still cause omissions, wrong alignment, or incomplete translation.

A further challenge is the presence of **rare philosophical, Vedic, and cultural terms**. Words such as *ātman*, *brahman*, *dharma*, *ṛta*, *mokṣa*, and *yajña* do not always have direct English equivalents. Their meaning depends on the philosophical school, textual tradition, and sentence context. If such words appear rarely in the training corpus, the model may translate them inconsistently or choose simplified English equivalents.

Limitations

The first limitation of the proposed framework is its dependence on the accuracy of the Sandhi splitting module. Since Sandhi splitting is performed before tokenization, any incorrect split may propagate through the entire translation pipeline. This is especially problematic in cases where multiple valid Sandhi splits are possible.

The second limitation is related to the use of **synthetic data**. Synthetic Sanskrit–English sentence pairs increase the size of the training corpus and help the model learn basic sentence patterns. However,

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

they may not fully represent the richness of real Sanskrit literature, such as poetic style, philosophical argumentation, Vedic expressions, and commentary-based writing. Therefore, results obtained using synthetic data should be validated further on large real Sanskrit–English corpora.

The third limitation is the handling of cultural and philosophical meanings. Many Sanskrit terms carry meanings that depend on context, tradition, and interpretation. A purely neural translation model may not fully capture this depth unless it is supported by external knowledge sources.

The fourth limitation is computational cost. Transformer-based models require more memory and training time than traditional models. This may limit their practical use in institutions with limited computational infrastructure.

Implications for Future Improvement

This error analysis suggests that future work should focus on improving contextual Sandhi disambiguation, compound interpretation, and culturally grounded translation. One promising direction is the integration of **Paninian grammar-based constraints** into the pre-processing and decoding stages. Paninian grammar can help the model better understand morphological structure, case relations, and syntactic dependencies.

Another important direction is **retrieval-augmented translation**. By connecting the model with Sanskrit dictionaries, commentaries, ontologies, and knowledge bases, the system can retrieve contextual meanings of rare philosophical and cultural terms during translation. This would help reduce literal translation errors and improve semantic accuracy.

6. Conclusion and Future Work

Conclusion

This study presented a **Sandhi-Aware Transformer Framework for Sanskrit Language Understanding and Machine Translation**. The work was motivated by the need for a translation system that can process Sanskrit according to its linguistic structure rather than treating it only as a generic low-resource language. The proposed framework combines Sanskrit-aware pre-processing, Sandhi splitting, byte-level tokenization, Transformer-based sequence modelling, and attention-based interpretation within a single translation pipeline.

The experimental results demonstrate that the proposed model performs better than RNN, LSTM, Seq2Seq with attention, and standard Transformer baselines. The improvement in BLEU score, ROUGE-L, token-level accuracy, and perplexity shows that Sanskrit-specific pre-processing contributes positively

The Brāhmī, International Multidisciplinary Research Journal
(Peer Reviewed, Referred & Open Access Journal)
Volume: 03 | Issue: 01 | Mar - May, 2026 | ISSN: 3048-9660 (Online) | www.thebrahmi.com

to translation quality. The ablation results further confirm that Sandhi splitting, byte-level tokenization, data augmentation, and transfer learning are useful components of the overall framework.

The main contribution of this work lies in integrating traditional Sanskrit linguistic knowledge with modern Transformer-based deep learning. This makes the framework suitable for Sanskrit–English translation, digital text processing, Sanskrit learning support, and computational access to Indian Knowledge Systems.

Future Work

Future research can extend this work in several directions. First, the model can be trained on larger and more diverse Sanskrit–English parallel corpora covering domains such as Vedic literature, epics, philosophy, Ayurveda, astronomy, grammar, and classical poetry. This would improve the system's ability to handle domain-specific vocabulary and complex sentence structures.

Second, more advanced Sandhi and compound disambiguation methods can be developed to resolve cases where a Sanskrit expression has multiple possible segmentations. The integration of Paninian grammar rules with neural models may further improve grammatical correctness and semantic consistency.

Third, retrieval-augmented translation can be explored by connecting the model with Sanskrit dictionaries, commentaries, and knowledge bases. This would be useful for translating culturally rich, philosophical, and technical terms.

Finally, the framework can be extended beyond Sanskrit–English translation to Sanskrit–Indian language translation, manuscript OCR-based translation, and speech-based Sanskrit applications. Developing open datasets, benchmark tasks, trained models, and evaluation tools will also support reproducible research in Sanskrit NLP and Indian Knowledge Systems.

References:

1. G. Cardona, *Pāṇini: A Survey of Research*. Delhi, India: Motilal Banarsidass, 1997.
2. P. Scharf and M. Hyman, *Linguistic Issues in Encoding Sanskrit*. Providence, RI, USA: Brown University, 2011.
3. S. Dave, A. K. Singh, P. A. Prathosh, and B. Lall, "Neural Compound-Word (Sandhi) Generation and Splitting in Sanskrit Language," arXiv preprint arXiv:2010.12940, 2020.
4. J. Sandhan, R. Singha, N. Rao, S. Samanta, L. Behera, and P. Goyal, "TransLIST: A Transformer-Based Linguistically Informed Sanskrit Tokenizer," arXiv preprint arXiv:2210.11753, 2022.

5. A. Vaswani et al., "Attention is All You Need," in *Advances in Neural Information Processing Systems*, 2017.
6. S. Nehrdich, O. Hellwig, and K. Keutzer, "One Model is All You Need: ByT5-Sanskrit, a Unified Model for Sanskrit NLP Tasks," arXiv preprint arXiv:2409.13920, 2024.
7. Y. Liu et al., "Multilingual Denoising Pre-training for Neural Machine Translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.
8. S. Khanuja et al., "MuRIL: Multilingual Representations for Indian Languages," arXiv preprint arXiv:2103.10730, 2021.
9. S. Doddapaneni et al., "Towards Leaving No Indic Language Behind: Building Monolingual Corpora, Benchmark and Models for Indic Languages," arXiv preprint arXiv:2212.05409, 2022.
10. K. Bhatt, K. N. J., G. Ramakrishnan, and P. Jyothi, "CharSS: Character-Level Transformer Model for Sanskrit Word Segmentation," arXiv preprint arXiv:2407.06331, 2024.
11. J. Sandhan, A. Agarwal, L. Behera, T. Sandhan, and P. Goyal, "SanskritShala: A Neural Sanskrit NLP Toolkit with Web-Based Interface for Pedagogical and Annotation Purposes," arXiv preprint arXiv:2302.09527, 2023.
12. M. B. Jagadeeshan, P. Raj, and P. Goyal, "Anveshana: A New Benchmark Dataset for Cross-Lingual Information Retrieval on English Queries and Sanskrit Documents," arXiv preprint arXiv:2505.19494, 2025.
13. S. Nehrdich, D. Allport, S. Sellmer, J. Sandhan, M. B. Jagadeeshan, P. Goyal, S. Kumar, and K. Keutzer, "Mitrasamgraha: A Comprehensive Classical Sanskrit Machine Translation Dataset," arXiv preprint arXiv:2601.07314, 2026.
14. B. Sadhukhan and S. Punyeshwarananda, "Automatic Speech Recognition for Sanskrit with Transfer Learning," arXiv preprint arXiv:2501.10024, 2025.