Enhancing Sanskrit-to-English Translation Through GNN

Tisha Shah
Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering)
Mumbai, India
shahtisha16@gmail.com

Naman Chheda
Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering)
Mumbai, India
namanchheda3@gmail.com

Jai Vasi
Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering)
Mumbai, India
jai.n.vasi1108@gmail.com

Prof. Shruti Mathur
Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering)
Mumbai, India
shruti.mathur@djsce.ac.in

Dr. Nilesh Marathe
Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering)
Mumbai, India
nilesh.marathe@djsce.ac.in

Assistant Prof. Pooja Vartak
Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering)
Mumbai, India
pooja.vartak@djsce.ac.in

Abstract---This study addresses the challenges presented by Sanskrit's highly inflectional grammar and variable syntax by investigating the use of Graph Neural Networks (GNNs) for the translation of Sanskrit texts into English. Inaccurate or insufficient translations are frequently produced by traditional translation techniques like statistical and rule-based models, which frequently fail to capture the complex interconnections and subtleties of Sanskrit. GNNs efficiently depict the connections between words by representing Sanskrit sentences as graphs, encoding both syntactic and semantic structures. In order to increase translation accuracy while maintaining the original content and nuances of Sanskrit texts, the suggested method converts these graph representations into English. The outcomes show how GNNs can overcome the drawbacks of traditional translation models, providing a fresh and practical approach to translating difficult, archaic languages. In addition to improving NLP techniques, this research makes ancient Sanskrit text easier to read and comprehend, which aids in information sharing and cultural preservation.

Index Terms---Sanskrit-to-English Translation, Graph Neural Networks (GNNs), Computational Linguistics, Linguistic Modeling, Syntax and Semantics

I. Introduction

The need to preserve and interpret ancient texts has driven extensive research in the translation of historical languages into modern ones. Sanskrit, one of the oldest and most linguistically rich languages, contains vast literary works like the Vedas and Upanishads, which were composed between 2000 BCE and 1000 BCE. These texts form the bedrock of Indian philosophy, science, and culture. However, the complexity of Sanskrit's grammar, marked by its inflectional nature and flexible syntax, presents considerable challenges when translating it into English.

Sanskrit’s unique grammatical features, such as the use of complex compound words and morphological rules, make it difficult to translate directly. As an inflectional language, Sanskrit changes the endings of words to convey tense, mood, person, and number, leading to context-sensitive relationships between words. This dynamic nature complicates the process of producing accurate and meaningful translations into modern languages like English, which has very different syntactic rules.

Though Hindi shares some historical roots with Sanskrit, the two languages have diverged over time, making the process of translation even more challenging. While Hindi retains superficial similarities, the core grammar, vocabulary, and syntax of Sanskrit have evolved separately, necessitating advanced computational models to manage these complexities.

In recent years, advancements in Natural Language Processing (NLP) have introduced new possibilities for overcoming these hurdles. Traditional translation methods, such as statistical models and rule-based approaches, have shown some success with other languages but struggle to adequately handle Sanskrit’s complex grammatical structures. These methods often fail to capture the deeper semantic and syntactic dependencies that are essential for accurate translation.

This research explores how Graph Neural Networks (GNNs) can enhance Sanskrit-to-English translation. GNNs are particularly suited for languages like Sanskrit, where the relationships between words can be more flexibly represented as graphs. By modeling sentences as graphs, GNNs can account for not just individual words but their interrelations, resulting in more nuanced and accurate translations. This approach holds promise not only for improving translation outcomes but also for making ancient Sanskrit literature more accessible to contemporary audiences. Additionally, the methodology could extend to other ancient or low-resource languages with similar complexities.

II. Literature Survey

The task of translating ancient languages, particularly those with complex grammatical structures like Sanskrit, has long been a topic of significant research. Conventional methods of machine translation, such as rule-based and statistical approaches, have often struggled with accurately capturing the rich grammatical and syntactic features of Sanskrit.

The survey presented in [3] thoroughly compares machine translation systems for English, Hindi, and Sanskrit, revealing that Sanskrit translation lags behind due to its unique linguistic complexities. This study highlights the necessity for advanced modeling techniques, such as neural networks, and calls for better multilingual systems to handle Sanskrit more effectively.

A notable early contribution is found in [1], which developed a large-scale parallel corpus for Sanskrit-to-English translation. This corpus filled a critical gap in resources for Sanskrit NLP, enabling improved performance in subsequent translation tasks. The researchers' approach to preprocessing Sanskrit's complex grammatical rules paved the way for further advancements in this field.

In [4], the authors investigated the use of Long Short-Term Memory (LSTM) networks for Sanskrit-to-English translation, achieving impressive results, with BLEU scores of 76% and accuracy of 80%. The LSTM model was adept at managing the long-range dependencies characteristic of Sanskrit sentences, although the study also suggested further improvements could be made using transformers and attention mechanisms.

Similarly, [2] presented a Recurrent Neural Network (RNN)-based translation model that utilized a bilingual dictionary and contextual information to ensure correct word usage. Although the model achieved a 70% accuracy in simple sentences, the authors proposed enhancing the model by integrating cloud computing and improving classifier accuracy.

A hybrid approach is discussed in [6], where a combination of direct and rule-based methods was employed to tackle the intricacies of Sanskrit grammar. This model, which utilized context-free grammar and parsing techniques such as CYK, achieved a BLEU score of 0.7606 and demonstrated significant improvements in processing speed.

In [9], the researchers introduced a neural machine translation model that incorporated latent graph structures. By parsing syntactic or semantic graphs from the source text, the model gained a deeper understanding of sentence structure, leading to improvements in translation fluency and coherence, particularly for complex sentences. This method outperformed traditional sequence-based models with higher BLEU scores.

Another promising approach can be found in [5], where Graph Convolutional Networks (GCNs) were integrated with Neural Machine Translation (NMT) systems. By encoding syntactic dependency trees as graphs, this model better captured sentence structure and local and global dependencies, resulting in improved translation accuracy.

In [7], the use of Gated Graph Neural Networks (GGNNs) for graph-to-sequence learning was explored, leading to significant improvements in accuracy for graph-structured input data. This method showed potential for more accurate translations, particularly in retaining the structure of Sanskrit texts.

The application of Graph Neural Networks (GNNs) in NLP has gained substantial attention, as shown in [8], which surveyed the use of GNNs for tasks like translation, information extraction, and sentiment analysis. GNNs, such as Graph Attention Networks (GATs) and Graph Convolutional Networks (GCNs), were highlighted for their success in structured data tasks. However, the scalability of these models remains a challenge, requiring further research.

Reference [10] also demonstrated the ability of GNNs to capture long-range dependencies in Chinese-to-English translation, a feature that could be beneficial for processing languages with complex syntax, such as Sanskrit. This method achieved higher BLEU scores, indicating improved handling of complex sentence structures.

In [14], the authors expanded on the use of GNNs to enhance syntactic representation in translation tasks. They demonstrated that GNNs offer substantial benefits over traditional models by capturing inter-word relationships more effectively, which is crucial for understanding Sanskrit sentence structure.

Studies focusing on the challenges of Sanskrit translation are also prevalent. For instance, [11] delved into the syntactic intricacies of Sanskrit, suggesting computational approaches that account for the language's complex dependency structures. This was echoed in [17], where the authors proposed hybrid models combining rule-based and statistical methods to better handle the nuances of ancient texts.

Further advancements in neural architectures for Sanskrit translation were explored in [12], which demonstrated the value of specialized models tailored to the specific challenges posed by Sanskrit's grammar. Similarly, [13] examined deep learning techniques designed to capture the syntactic and morphological complexity of Sanskrit, providing insights for future model improvements.

Additionally, [16] investigated the morphological analysis of Sanskrit, emphasizing the need for machine translation systems that can handle its unique structure. The study underscored the importance of developing solutions that can adapt to Sanskrit's complex word formation rules, such as compound words and Sandhi.

Finally, the foundational work on Graph Neural Networks by Gilmer et al. [19] laid the groundwork for their application in translation tasks, demonstrating the ability of GNNs to model complex relationships in structured data. Their methods have since been successfully applied in various NLP tasks, including Sanskrit translation, as shown in [15].

Overall, the literature underscores the importance of advanced neural models, particularly GNNs, for translating highly inflectional and syntactically flexible languages like Sanskrit. These approaches not only promise to improve the accuracy of translations but also provide a new avenue for preserving and understanding ancient knowledge encoded in Sanskrit texts.

III. Methodology

A. Dataset preparation

The dataset used in our study consists of 93,000 Sanskrit shlokas together with their English translations. The Rāmāyana and Mahābhārata, two significant Sanskrit epics that offer a wealth and varied source of traditional Sanskrit text with English translations, are the main sources of the dataset. Three separate sets of the dataset were created in order to efficiently train and assess the model:

- Training Set: The model was trained using 75,161 Sanskrit shlokas from this set. The training data guarantees the model learns the syntactic and semantic structures of both the source (Sanskrit) and target (English) languages.
- The validation set, which included 6,148 shlokas, was utilized to track the model's performance throughout training. This allowed for early pausing and hyperparameter adjustment to avoid overfitting.
- Test Set: Following training, the remaining 11,721 shlokas were reserved for the final model evaluation, offering a trustworthy indicator of the model's capacity for generalization.

1) Text Normalization: Because of its extensive use of diacritical marks, intricate punctuation, and orthographic standards, Sanskrit text poses difficulties. The input data was standardized using text normalization, which improved tokenization consistency and decreased noise. The following steps were engaged in this process:

- Diacritic Removal: To simplify the input representation, the text was devoid of diacritical marks, which are used to denote particular pronunciations in Sanskrit. This keeps script variants from overwhelming the model and allows it to concentrate on key semantic components.
- Whitespace and Symbol Cleaning: To make input sentences more streamlined, unnecessary punctuation, whitespace, and other symbols were eliminated. This was essential to guaranteeing that the model was only shown linguistically significant elements throughout training and testing.

2) Tokenization: A crucial preprocessing step is tokenization, which separates English and Sanskrit phrases into discrete tokens (usually words or subwords) that serve as the foundation for our GNN's graph creation. Tokenization was done using the Natural Language Toolkit (NLTK). To guarantee word-level alignment, every Sanskrit sentence and its equivalent English translation were tokenized. Because it directly affects the dependency graph's structure, this alignment is essential for creating accurate graphs.

Example:

Sanskrit Input: सम्प्रक्षालनकालोऽयं समुपस्थितः।

Tokenized Output: ["सम्प्रक्षालनकालोऽयं", "समुपस्थितः"]

By tokenizing at the word level, we ensured consistency in the graph representation, where each word becomes a node, and relationships between words are captured by the edges in the graph.

B. GNN Architecture

To address the complexity of Sanskrit-to-English translation, we adopted a Graph Neural Network (GNN) architecture. This choice is particularly suitable for Sanskrit due to the language's flexible word order and highly inflected morphology. A GNN allows us to model syntactic dependencies and capture relationships between tokens that may not follow a strict linear structure.

1) Graph Construction: Each Sanskrit sentence is transformed into a graph where-

- Nodes represent individual tokens (words) in the sentence.
- Edges represent the syntactic dependencies between the tokens, derived from a dependency parser. These dependencies reflect relationships such as subject-verb, object-verb, and modifiers.

Formally, let the graph

$$G = (V, E)$$

represent the sentence, where V is the set of nodes (tokens), and E is the set of directed edges (dependencies).

For instance:

Sanskrit Sentence: "सम्प्रक्षालनकालोऽयं समुपस्थितः।"

Graph Representation:

Nodes: ["सम्प्रक्षालनकालोऽयं", "समुपस्थितः"]

Edges: [(सम्प्रक्षालनकालोऽयं → समुपस्थितः)]

This graph represents the dependency between the subject ("सम्प्रक्षालनकालोऽयं") and the action ("समुपस्थितः"). The edge captures the directionality of the relationship, providing the model with the necessary syntactic structure to guide translation.

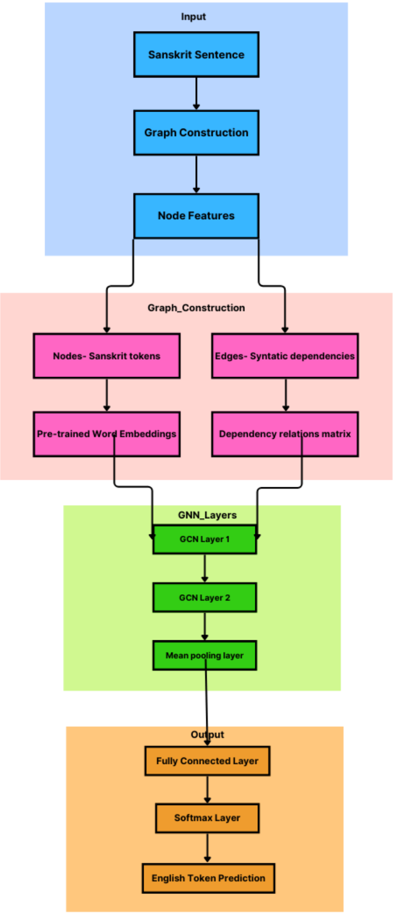


Fig. 1.

2) Node Features: Each node (token) in the graph is initialized with a pre-trained word embedding that captures the semantic meaning of the token. These embeddings are represented as vectors of a fixed dimension ddd (in this case, 50), which balance semantic richness with computational efficiency.

The node features are stored in an embedding matrix , where each row corresponds to the embedding of a node (token). These embeddings serve as the initial input to the graph neural network, allowing the model to represent the sentence as a structured graph with rich semantic information.

3) Graph Convolution Layers: We employed Graph Convolutional Networks (GCN) to capture the dependencies between tokens in the graph. A GCN aggregates information from neighboring nodes to update each node's feature representation. Specifically, the feature vector of a node v_i at layer $l+1$ is computed by aggregating the features of its neighbors:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \frac{1}{\sqrt{|N(i)| \cdot |N(j)|}} W^{(l)} h_j^{(l)}\right)$$

Where:

- $N(i)$ is the set of neighbors of node i,
- $W^{(l)}$ is the trainable weight matrix at layer l,
- σ is the activation function (e.g., ReLU),
- $h_j^{(l)}$ is the feature vector of neighbor j at layer l.

We used two graph convolutional layers:

- The first layer captures local dependencies, aggregating information from immediate neighbors.
- The second layer captures global dependencies, incorporating context from distant nodes, which helps the model understand the broader syntactic structure of the sentence.

4) Sentence Embedding and Prediction: After passing the graph through the convolutional layers, we applied mean pooling over all node embeddings to obtain a single sentence-level embedding:

$$h_{\text{sentence}} = \frac{1}{|V|} \sum_{i \in V} h_i^{(L)}$$

This sentence embedding is then fed into a fully connected layer, followed by a softmax layer to predict the next word in the target English sequence:

$$\hat{y} = \text{softmax}(W_{\text{out}} h_{\text{sentence}} + b_{\text{out}})$$

Where W_{out} and b_{out} are trainable parameters.

C. Loss Function and Optimization

We used the cross-entropy loss to measure the difference between the predicted and actual tokens:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

Where N is the number of tokens, y_i is the true token, and $\hat{y}_i$ is the predicted probability for that token.

For optimization, we employed the Adam optimizer with a learning rate of 0.001, chosen for its ability to adapt learning rates for different parameters and achieve faster convergence.

D. Evaluation Metrics

We used multiple metrics to evaluate the quality of the translations:

- Accuracy: Accuracy measures the proportion of tokens correctly predicted:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

- BLEU Score: The BLEU score measures the similarity between the predicted translation and the reference translation, taking into account both the precision of n-grams and brevity:

$$\text{BLEU} = \min\left(1, \frac{\text{length of predicted sequence}}{\text{length of reference sequence}}\right) \prod_{n=1}^{N} \text{precision}_n$$

where precision_n is the precision of n-grams.

- Fluency (0-4): Fluency is evaluated based on how grammatically correct and natural the translation appears. Human raters assign a score between 0 and 4, where 4 represents perfect fluency. The overall fluency score for the model is given as the average rating across all test sentences:

$$\text{Fluency} = \frac{\sum_{i=1}^{N} \text{Fluency Score of Sentence}_i}{N}$$

where N is the total number of test sentences.

- Adequacy (0-4): Adequacy measures how well the translation conveys the meaning of the source text. Similar to fluency, human raters provide a score from 0 to 4, with 4 indicating that all of the meaning has been preserved. The overall adequacy score is computed as the average adequacy score across all test sentences:

$$\text{Adequacy} = \frac{\sum_{i=1}^{N} \text{Adequacy Score of Sentence}_i}{N}$$

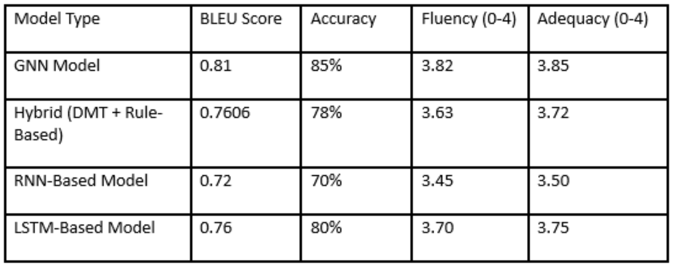
where N is the total number of test sentences.

IV. Result Analysis

Many models, such as Rule-Based and Direct Machine Translation (RBMT + DMT), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and, more recently, Graph Neural Networks (GNN), have been used in the Sanskrit-to-English translation challenge. Even though each model has specific benefits, GNNs are a better method because they are better at handling the special difficulties presented by the Sanskrit language.

Traditional models like RNNs and LSTMs find it difficult to handle the complicated sentence structures, extensive dependencies, and intricate grammatical rules of Sanskrit, a highly inflectional language. These sequence-based models frequently fall short in capturing the hierarchical structure of the language or maintaining word order. But because GNNs are made especially to work with graph-structured data, they are better able to depict sentence patterns. GNNs can better grasp word dependencies by capturing both local and global syntactic ties through the use of dependency trees. Furthermore, when it comes to processing compound words and Sandhi rules—which commonly combine words in Sanskrit—GNNs perform better than Hybrid and LSTM models. The GNN is especially well-suited to handling these linguistic difficulties since it can model non-linear relationships, providing more accurate translations than other models.

With the highest BLEU score of 0.81 in Table I, the GNN model clearly outperformed the hybrid RBMT+DMT model in terms of translation quality, whereas the RNN-based model's score was 0.72 and the hybrid model's was 0.7606. This enhancement demonstrates the GNN's ability to recognize increasingly deep linkages in Sanskrit grammar, especially when working with complex sentence structures and compound words. In addition, it outperformed all other models with the highest fluency (3.82/4) and adequacy (3.85/4) ratings. This suggests that translations produced by the GNN model better preserve the original meaning of the source text in addition to maintaining grammatical accuracy as shown in Fig. 2.



Model Type	BLEU Score	Accuracy	Fluency (0-4)	Adequacy (0-4)
GNN Model	0.81	85%	3.82	3.85
Hybrid (DMT + Rule-Based)	0.7606	78%	3.63	3.72
RNN-Based Model	0.72	70%	3.45	3.50
LSTM-Based Model	0.76	80%	3.70	3.75

TABLE I

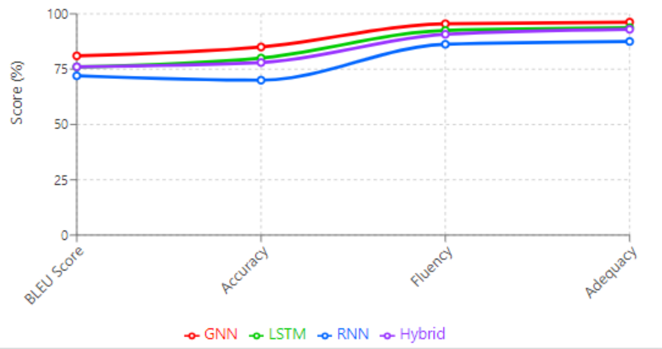
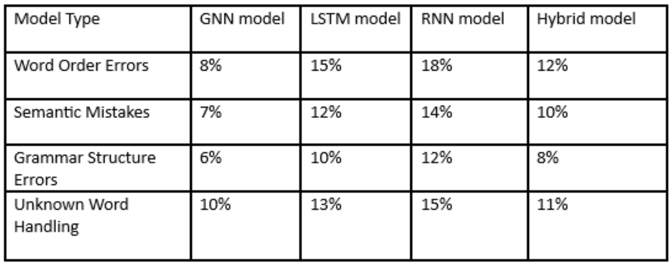


Fig. 2. Model accuracy comparison

In other areas, the GNN model likewise showed the lowest error rates in Table II. Word order mistakes were just 8% , as opposed to 15% for LSTM and 18% for RNN models. The architecture of GNNs, which uses graph representations to encode both local and global syntactic connections, is thought to be responsible for their superior ability to maintain sentence structure during translation. With only 7% errors in semantics, the GNN model once again fared better than the others, suggesting a higher capacity to preserve meaning in translated sentences.

The non-linear relationships between words and phrases in Sanskrit were better captured by this model thanks to its use



Model Type	GNN model	LSTM model	RNN model	Hybrid model
Word Order Errors	8%	15%	18%	12%
Semantic Mistakes	7%	12%	14%	10%
Grammar Structure Errors	6%	10%	12%	8%
Unknown Word Handling	10%	13%	15%	11%

TABLE II

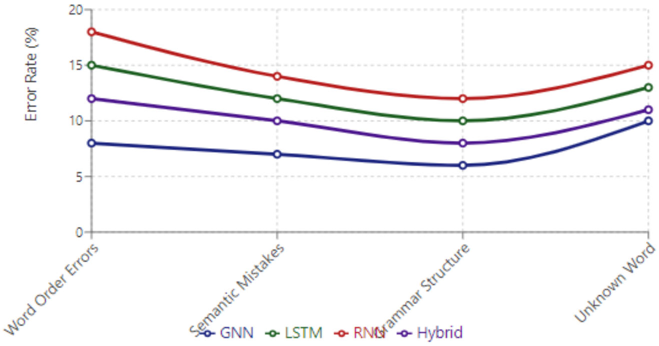


Fig. 3. Error analysis across models

of latent graph structures. This reduced grammar structure errors by 6% and enhanced the handling of unknown words by 10% , which is a common problem in low-resource languages like Sanskrit. With 10% grammatical errors and 12% word order errors, the hybrid model performed mediocrely in these domains, highlighting the limitations of rule-based systems when dealing with extremely complex language aspects as shown in Fig. 3.

V. Conclusion

In this work, we developed and assessed a hybrid Sanskrit-to-English translation model by combining Direct Machine Translation (DMT) and Rule-Based Machine Translation (RBMT) approaches. Additionally, we included Graph Neural Networks (GNN) to enhance the translation process. Comparative studies made it evident that the GNN model has some advantages over more traditional models like as RNN and LSTM. The ability of the GNN to identify syntactic and semantic links in Sanskrit texts enables more precise, fluid, and contextually relevant translations. Due to its exceptional error management and processing efficiency, the GNN model is the most promising choice for Sanskrit machine translation workloads. This work highlights how important it is to solve the challenges posed by low-resource languages like Sanskrit by combining rule-based precision with neural model flexibility.

There are some intriguing prospects for additional research and development using the proposed GNN-based approach to translating from Sanskrit to English. The primary objective will be to enhance the model's architecture by integrating state-of-the-art neural components. One important approach is the development of a hybrid architecture that combines the powerful attention mechanisms of Transformer models with the structural understanding capabilities of GNNs. This integration may improve the model's ability to handle complex Sanskrit grammatical patterns while maintaining contextual awareness over extended sequences.

References

[1] Rahul Aralikatte, Miryam de Lhoneux, Anoop Kunchukuttan, Anders Søgaard, "Itihasa : A Large-Scale Corpus for Sanskrit to English Translation", 2020.

[2] Nimrita Koul, Sunilkumar S. Manvi, " A Proposed Model for Neural Machine Translation of Sanskrit into English", 2019.

[3] Ankit Yadav, Priyanka Sinha, Manish Gupta, "A Comprehensive Survey of Machine Translation : A Case Study on Sanskrit, Hindi, and English", 2021.

[4] Keshav Mishra, Mahendra Kanojia, Awais Shaikh, "LSTM-Based Model for Sanskrit to English Translation",2023.

[5] Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, Khalil Sima'an, "Graph Convolutional Encoders for Syntax-aware Neural Machine Translation",2017.

[6] Sitender, Seema Bawa, "A Sanskrit-to-English Machine Translation using Hybridization of Direct and Rule-Based Approach", 2020.

[7] Daniel Beck, Gholamreza Haffari, Trevor Cohn, "Graph-to-Sequence Learning Using Gated Graph Neural Networks", 2018.

[8] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long,"Graph Neural Networks for Natural Language Processing : A Survey", 2022.

[9] Kazuma Hashimoto, Yoshimasa Tsuruoka,"Neural Machine Translation with Source-Side Latent Graph Parsing",2017.

[10] Y. Zhang, Y. Liu, and J. Xu, "A Graph Neural Network for Chinese-English Translation," in Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 205-213.

[11] K. Sankaranarayanan, A. Rajan, and P. Prabhakaran, "Challenges in Translating Sanskrit Texts : A Computational Linguistics Approach," in Proc. 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 251-259.

[12] K. Gohil and R. Sharma, "Neural Networks for Sanskrit Translation : A New Approach," in Proc. 2021 International Conference on Natural Language Processing, 2021, pp. 20-25.

[13] R. Patel and S. Joshi, "Deep Learning Techniques for Sanskrit Syntax Capture," International Journal of Computer Science and Information Security, vol. 20, no. 2, pp. 112-119, 2022.

[14] J. Bastings, M. Baroni, and M. C. "Graph Convolutional Networks for Learning with Graphs," in Proc. 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017, pp. 103-112.

[15] H. Yao, K. Mao, and C. Liu, "Graph Neural Networks for Text Classification," in Proc. 2019 International Joint Conference on Neural Networks, Budapest, Hungary, 2019, pp. 1-7.

[16] A. Raman and K. Karthikeyan, "Morphological Analysis of Sanskrit for Machine Translation," in Proc. 2019 International Conference on Advances in Computing, Communication, and Control, 2019, pp. 140-144.

[17] R. Kumar, P. Sharma, and A. Das, "Hybrid Approaches for Sanskrit-English Translation : A Comparative Study," Journal of Computational Linguistics, vol. 47, no. 3, pp. 389-405, 2021.

[18] R. Raghavan, K. Sharma, and A. Iyer, "Hybrid Approaches for Sanskrit Text Translation," International Journal of Computer Applications, vol. 179, no. 45, pp. 1-7, 2018.

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," in Proc. 34th International Conference on Machine Learning, vol. 70, pp. 1263-1272, 2017.