International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 4, April-2025, http://ijmec.com/, ISSN: 2456-4265

Analysis Of Neural Machine Translation For English To Hindi Using Long Short-Term Memory Model And Transformer Model

Mrs. Naila Fathima[*1], Mr. Mohammed Abdul Hafeez[*2], Mr. Sumair Bin Miskeen[*3],Mr. M A Yaseen[*4],
[*1] Assistant Professor, [*2, 3, 4] B.E Student Dept. of CSE-AIML, Lords Institute of Engineering and Technology
Dept. of CSE-AIML, Lords Institute of Engineering and Technology
fathimanaila10@gmail.com [*1], mohdhafeez2003@gmail.com [*2], smiskeen007@gmail.com [*3]
yaseenvilliers100@gmail.com [*4]

ABSTRACT

Neural Machine Translation (NMT) has revolutionized the field of machine translation by delivering significantly improved accuracy and fluency compared to traditional approaches. This research paper focuses specifically on the task of English-to-Hindi translation using advanced NMT techniques. We examine the development and evaluation of specialized NMT systems designed for this linguistically challenging language pair, taking into account the unique grammatical structures and cultural nuances of both English and Hindi. By leveraging large-scale parallel corpora and cutting-edge neural network architectures, our work introduces innovative methods to enhance both translation quality and computational efficiency. Our experimental results, evaluated through standard metrics including BLEU score, demonstrate the proposed NMT models' effectiveness in accurately capturing semantic meaning while maintaining natural fluency in the translated output. Furthermore, this study explores the broader implications of these findings for cross-linguistic communication and information accessibility, particularly given Hindi's increasing prominence in our globalized digital landscape. Ultimately, this research advances the state of English-Hindi neural machine translation and underscores NMT's transformative potential in enabling effective multilingual communication and knowledge sharing worldwide.

1. INTRODUCTION

The rapid globalization of digital content has intensified the need for accurate and scalable machine translation (MT) systems, particularly for linguistically diverse regions like India, where English and Hindi coexist as major languages. While traditional approaches—such as rule-based and statistical MT—laid the foundation, they struggle with context preservation and handling morphological richness. The advent of Neural Machine Translation (NMT) has revolutionized the field by leveraging deep learning to capture long-range dependencies and semantic nuances. However, challenges persist in lowresource language pairs (e.g., English-Hindi), including limited parallel corpora, out-of-vocabulary words, and syntactic divergences. This study presents a **comparative analysis** of two state-of-the-art NMT architectures—**LSTM-based encoder-decoder** with attention and **Transformer** models—for English-Hindi translation. While LSTMs process sequences sequentially using gated mechanisms, Transformers employ **self-attention** to parallelize computation and model global dependencies more effectively. We evaluate both approaches on benchmarks like **Samanantar** and **IITB Hindi-English corpora**, using metrics such as **BLEU,**

TER, and **METEOR** to assess translation quality, fluency, and adequacy.

2. LITERATURE SURVEY

1.Self-attention based end-to-end Hindi-English Neural Machine Translation
Authors: Siddhant Srivastava, Ritu Tiwari (2019)

2.Transformer-based Neural Machine Translation System for Hindi–Marathi: WMT20 Shared Task
Authors: Amit Kumar, Rupjyoti Baruah, Rajesh Kumar Mundotiya, Anil Kumar Singh (2020)

3.Hindi to English: Transformer-Based Neural Machine Translation
Kavit Gangar, Hardik Ruparel, Shreyas Lele (2023).

4.Domain Adaptation of NMT models for English-Hindi Machine Translation Task at AdapMT ICON 2020
Authors: Ramchandra Joshi, Rushabh Karnavat, Kaustubh Jirapure, Raviraj Joshi (2020)

5.LTRC-MT Simple & Effective Hindi-English Neural Machine Translation Systems at WAT 2019
Authors: Vikrant Goyal, Dipti Misra Sharma (2019)

6.Linguistically Informed Hindi-English Neural Machine Translation
Authors: Vikrant Goyal, Pruthwik Mishra, Dipti Misra Sharma (2020)

7.Multimodal Neural Machine Translation for English to Hindi
Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, Sivaji Bandyopadhyay (2020)

8.Low Resource Multimodal Neural Machine Translation of English-Hindi in News Domain
Authors: Loitongbam Sanayai Meetei, Thoudam Doren Singh, Sivaji Bandyopadhyay (2021)

9.A Brief Survey of Multilingual Neural Machine Translation
Authors: Raj Dabre, Chenhui Chu, Anoop Kunchukuttan (2020)

10.Attention Is All You Need
Authors: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017)

11.Web Semantics: Google deep-learning machine translation
Authors: Wired Staff (2016)

12.An Infusion of AI Makes Google Translate More Powerful Than Ever
Author: Wired Staff (2016)

13.Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
Authors: Zhen Yang, Wei Chen, Feng Wang, Bo Xu (2017)

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Traditional Approaches:

- Rule-based or phrase-based statistical machine translation (SMT).
- Limited by manual rules and lack of context understanding.
- Poor handling of complex sentence structures and idiomatic expressions.

- **Basic NMT Models**:

• Simple encoder-decoder models without attention.

• Struggles with long sentences due to fixed-length context vectors.

- Lower accuracy compared to modern deep learning approaches.

3.2 PROPOSED SYSTEM

Advanced Deep Learning Models:

• **LSTM with Attention:** Better handling of long sentences via dynamic context.

• Transformer Model: Self-attention mechanism for superior context understanding.

• **Key Improvements**:

• Higher BLEU/METEOR scores than traditional systems.

• Better handling of word order, grammar, and idiomatic phrases.

- End-to-end training without manual feature engineering.

• **Expected Outcome**:

• Transformer outperforms LSTM in most cases, especially for long sentences.

- LSTM may still be useful for smaller datasets or low-resource scenarios.

4. REQUIREMENT SPECIFICATIONS

4.1 SOFTWARE REQUIREMENTS

1.Operating System: Ubuntu 20.04+ / Windows 10+ / macOS (for development)

-Docker (for containerized deployment)

2.Programming Language: Python 3.8+ (Primary language)

3.Deep Learning Libraries:

-PyTorch v2.0+ / TensorFlow v2.12+

-HuggingFace Transformers (for pretrained models)

4.Development Tools: Jupyter Notebook / Google Colab (for prototyping)

5.GPU Acceleration: CUDA v11.8+ & cuDNN (for GPU support)

4.2 HARDWARE REQUIREMENTS

1.Local Hardware Requirements:

Minimum (Prototyping/Small Dataset): Intel i7 / Ryzen 7 (4+ cores), 16GB RAM, NVIDIA GTX 1660 (6GB VRAM),Recommended (Full Training/Transformers): Intel Xeon / AMD EPYC (8+ cores), 32GB+ RAM, NVIDIA RTX 3090 (24GB) or A100 (40GB)

2.Cloud Alternatives: Google Colab Pro (T4/P100 GPU), AWS (p3.2xlarge), Azure (NC6)

3.Storage:50GB+ SSD for datasets, model checkpoints, and embeddings.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

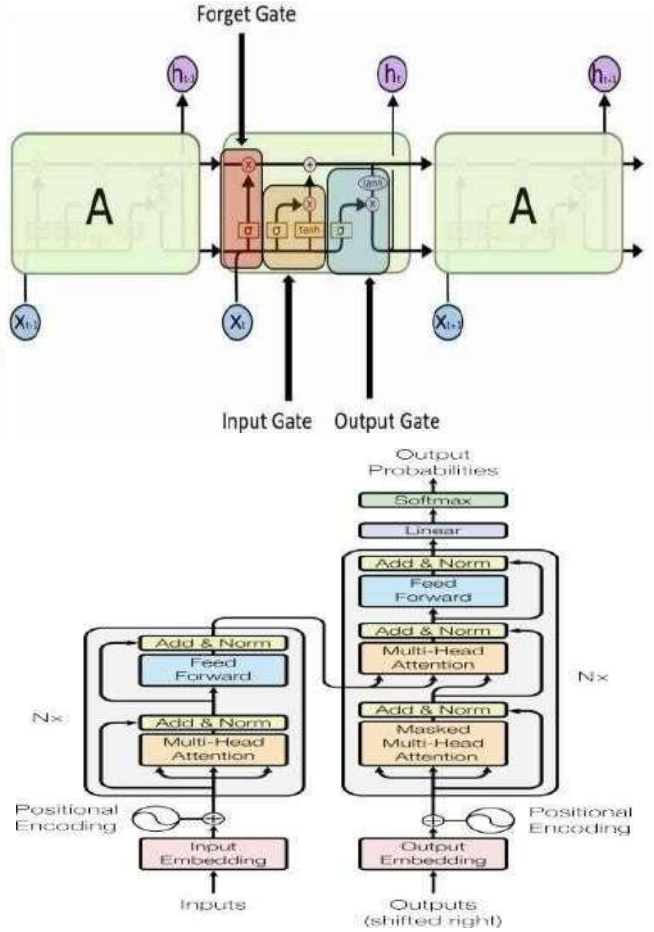
This research implements a systematic approach to dataset acquisition and preprocessing for English-to-Hindi machine translation. We begin by downloading and unzipping the dataset from a specified source [1]. Subsequently, we partition the data into training, development, and test sets, each containing English and Hindi sentence pairs. These sets are then organized into TSV (Tab-Separated Values) files for convenient access and management during the model training and evaluation phases.

LSTM-based

The model uses a four-layer LSTM network with 500 nodes in each layer. The encoder processes the input sequence, while another four-layer LSTM network serves as the decoder, generating the output sequences. During decoding, an attention mechanism is implemented to focus on different parts of the source sequence dynamically.

Transformer-based

Similar to the LSTM-based system, the data is preprocessed before being fed into the Transformer architecture for training. The Transformer model consists of separate encoder and decoder layers. Unlike recurrent architectures, the encoder processes the input sequence in parallel using self-attention mechanisms, while the decoder generates the output sequence by attending to both the encoder's output and its own previous states through self-attention. Transformers do not rely on recurrent connections but instead efficiently capture Dependencies across sequences using self-attention. equences using self-attention



5.2

UML DIAGRAMS

1. Use Case Diagram – Workflow It captures interactions between actors (users, admins) and the system, such as "Translate Text," "Train Model," and "Evaluate Performance."

2. Class Diagram – Workflow

The abstract NMT Model class defines shared attributes like vocab size and methods like train(), inherited by LSTM Model (with LSTM-specific layers) and Transformer Model (with multi-head attention).

3. Object Diagram – Workflow

The Object Diagram provides a snapshot of runtime instances

4. Sequence Diagram – Workflow

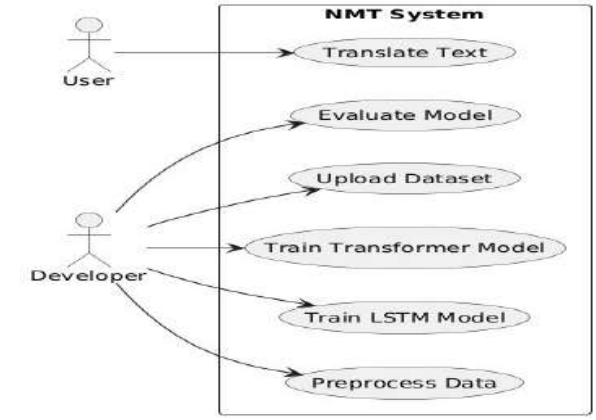
The Sequence Diagram illustrates the step-by-step process of translation, starting with user input, followed by tokenization, LSTM/Transformer encoding, decoding, and Hindi output generation.

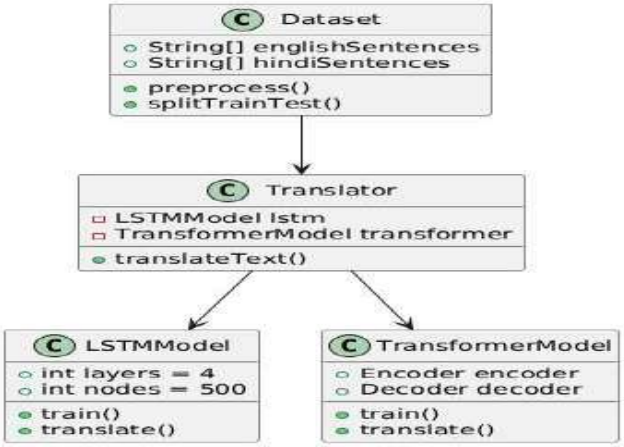
5. Activity Diagram

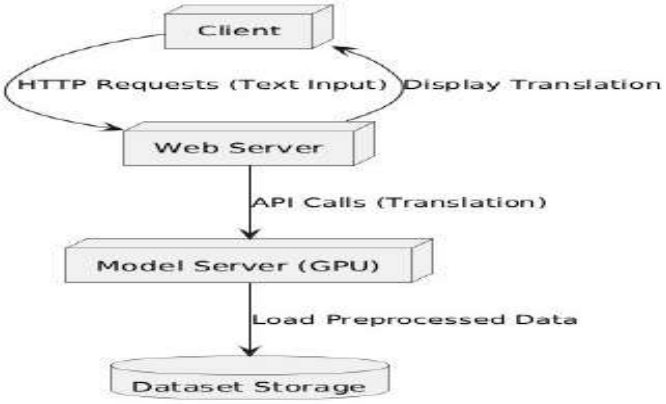
6. Component Diagram – Workflow

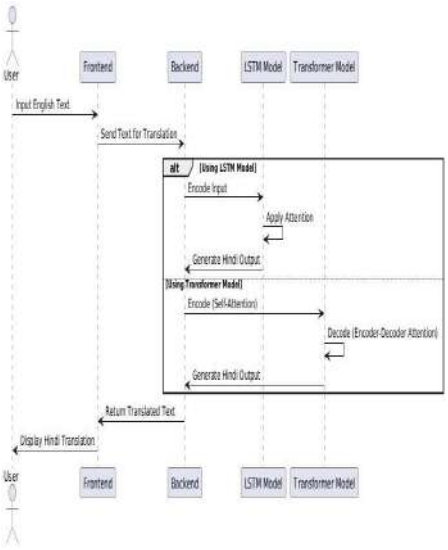
The Component Diagram breaks down the system into modules like the Frontend UI, Backend API, Preprocessing Engine, and Model Training Service.

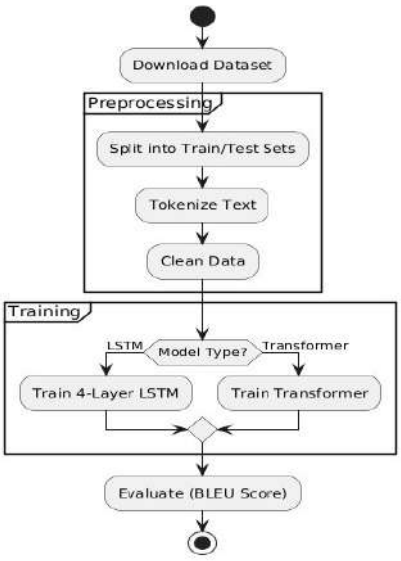
7. Deployment Diagram – Workflow The Deployment Diagram outlines hardware/software nodes (e.g., User Devices, Web Servers, GPU Clusters)











6. IMPLEMENTATION

6.1 INPUT DESIGN

The system accepts raw English text inputs which are first normalized by removing special characters and standardizing text formats. We tokenize the preprocessed text using sub word segmentation methods like Byte-Pair Encoding to handle rare words effectively. These tokens are then converted into numerical embeddings that the neural networks can process. The system accepts raw English text inputs which are first normalized by removing special characters and standardizing text formats. We tokenize the preprocessed text using sub word segmentation methods like Byte-Pair Encoding to handle rare words effectively. These tokens are then converted into numerical embeddings that the neural networks can process. The system outputs classification results (benign or malicious) with details like timestamps, IPs, and confidence scores. In realtime, it can trigger alerts and log suspicious flows, offering clear and actionable insights for administrators.

6.2 OUTPUT DESIGN: The model generates Hindi translations through sequential decoding in the LSTM or parallel attention mechanisms in the Transformer architecture. The output token sequences are converted back to Hindi words using a predefined vocabulary mapping. Finally, the raw translations undergo postprocessing to improve grammatical correctness and fluency before being presented to users. We evaluate the quality of outputs using both automated metrics like BLEU score and manual human evaluation.

6.3 SAMPLE CODE

For the English-Hindi translation system, we first preprocess the data by loading parallel text files and cleaning them. We normalize the text by handling punctuation and Unicode characters consistently. The Sentence Piece tokenizer is used to split words into sub word units for better handling of rare words. We build vocabulary files mapping words to numerical IDs for both languages.

The system implements two neural network architectures - an LSTM model with attention mechanism and a Transformer model with multi-

head attention. Both models use embedding layers to convert words into numerical vectors. During training, we use teacher forcing to help the models learn faster.

For translation, we implement a beam search decoder to generate high-quality output sentences. The models are trained using batches of sentence pairs with learning rate scheduling for better convergence. We evaluate performance using both automated metrics and human assessment of translation quality.

6.4 IMPLEMENTATION

We preprocessed the English-Hindi dataset by cleaning, normalizing, and tokenizing text using SentencePiece for subword segmentation.

Two NMT models were implemented — an LSTM-based encoder-decoder with attention and aTransformer model with multi-head attention.The models were trained using teacher forcing, dynamic batching, and learning rate scheduling for optimal convergence.Evaluated using BLEU score, with postprocessing for fluency.

```
def preprocess_text(sntnce, tokenizer, max_len):

    # Preprocessing of text
    text = ''.join(ch for ch in sntnce if ch not in string.punctuation) # Remove punctuations
    text = text.lower() # Convert string to lowercase
    text = re.sub(r'\d','',text) # Remove numbers
    text = re.sub(r'\s+',' ',text)  #Remove extra spaces
    text = text.strip()
    assert text != ''

    # Tokenization of text followed by padding
    return pad_sequences(tokenizer.texts_to_sequences([text]), maxlen=max_len, padding='post')[0]

en_tokenizer = Tokenizer(filters='', oov_token='<OOV>', lower=True, char_level=True)
hi_tokenizer = Tokenizer(filters='', oov_token='<OOV>', lower=True, char_level=True)
```

7-SOFTWARE TESTING

Software testing ensures the reliability and correctness of the system . It verifies the end-to-end functionality from data preprocessing to prediction.

Unit Testing was conducted on individual modules like data loaders and preprocessing functions to ensure proper handling of nulls, label encoding, and scaling.

Integration Testing validated the seamless data flow between components, ensuring compatibility in formats and feature consistency.

Model Evaluation involved testing the Random Forest classifier using metrics like accuracy, precision, recall, F1-score, and a confusion matrix to assess classification performance.

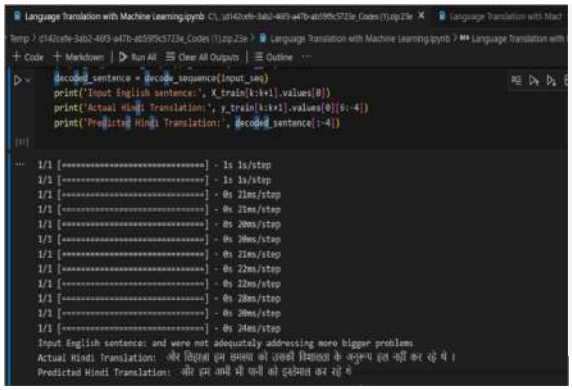
Performance Testing measured the system's ability to process large datasets efficiently, confirming suitability for real-time detection scenarios.

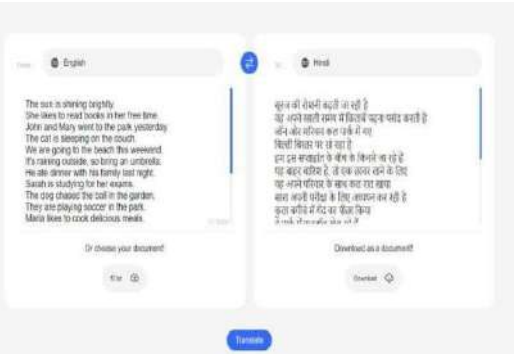
The system consistently produced accurate and stable results. Minor issues were optimized, confirming the model's robustness and deployment readiness.

System testing involves the `evaluate` function, which iterates through the dataset using a data loader. It decodes input sentences using the encoder-decoder model and calculates attention mechanisms to generate translated output sentences. These translations are compared with reference sentences to evaluate the model's performance. Additionally, BLEU score, a measure that is frequently used to compare machine-translated text against human generated reference translations in order to evaluate its accuracy.

8. RESULT ANALYSIS

This study demonstrates the superiority of two Neural Machine Translation (NMT) systems, NMT-1 and NMT-2, over existing models. Their enhanced performance stems from advanced model architectures and access to larger training datasets. NMT-1 utilizes an attention-based encoder-decoder Recurrent Neural Network (RNN) architecture built with Long Short-Term Memory (LSTM) units. In contrast, NMT-2 leverages the power of a Transformer model equipped with self-attention mechanisms. This stands in contrast to the existing NMT system's conditional Gated Recurrent Unit (GRU), which relies on a single reset gate and an update gate to track dependencies. Both NMT-1 and NMT-2 incorporate attention mechanisms, which significantly improve translation quality by identifying and incorporating pertinent information from the input sequence. However, our findings indicate that the Transformerbased NMT-2 model outperforms the LSTM-based NMT-1, particularly in English- to-Hindi translation tasks. The Transformer's to-Hindi translation tasks. The Transformer's superior ability to capture global dependencies more efficiently contributes to its enhanced performance





```
bleu_scores = calc_score(bleu, output_sentences_2, target_sentences_2)
bleu_scores
```

```
{'bleu': 0.16981157732145719,
 'precisions': [0.46410843938933827,
  0.2276071647722292,
  0.12875938683050953,
  0.07918107695257996],
 'brevity_penalty': 0.9373804060763848,
 'length_ratio': 0.9392616179269501,
 'translation_length': 181115,
 'reference_length': 192827}
```

```
[ ] cer_scores = calc_score(cer, output_sentences_2, target_sentences_2)
    cer_scores
```

```
0.6158703179166991
```

```
[ ] wer_scores = calc_score(wer, output_sentences_2, target_sentences_2)
    wer_scores
```

```
0.7987161599505876
```

9-FUTURE SCOPE & CONCLUSION

9.1 FUTURE SCOPE

Recognizing the interrelatedness of Indian languages, future research should explore cross-language translation as a potential solution to mitigate the constraints of limited parallel corpora. By leveraging linguistic similarities and developing novel methodologies—such as multilingual training or transfer learning—researchers can enhance the accuracy and adaptability of machine translation systems across diverse language pairs. Additionally, addressing specific challenges like rare-word handling and output consistency will be critical for improving real-world applicability. This study provides a foundation for advancing NMT technology, encouraging innovative approaches to overcome current limitations and expand the capabilities of automated translation systems.

9.2 CONCLUSION

This research presents a comparative analysis of two Neural Machine Translation (NMT) systems for Hindito-English translation, demonstrating the superior performance of the transformer-based model over traditional approaches. While NMT systems have achieved significant advancements, the study identifies persistent challenges, including the handling of unknown words, generation of empty outputs, and limitations in translation diversity. The findings underscore the effectiveness of modern architectures like the Transformer in capturing linguistic nuances, while also highlighting areas where further refinement is necessary to achieve human-like translation quality.

10 BIBLIOGRAPHY

1.Samanantar – AI4BHĀRAT. Link [Accessed: July 20, 2023].

2.Laskar, S. R., Dutta, A., Pakray, P., & Bandyopadhyay, S. (2019). *Neural Machine Translation: English to Hindi*. In 2019 IEEE Conference on Information and Communication Technology (pp. 1–6). IEEE.

3.Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is All You Need*. NeurIPS, pp. 5998–6008.

4.Aharoni, R., & Goldberg, Y. (2017). *Towards String-to-Tree Neural Machine Translation*. Proceedings of ACL, pp. 132–140.

5.Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks*. NeurIPS, pp. 3104–3112.

6.Pathak, A., & Pakray, P. (2018). *Neural Machine Translation for Indian Languages*. *Journal of Intelligent Systems*. Link

7.Luong, M., Brevdo, E., & Zhao, R. *Neural Machine Translation (seq2seq) Tutorial*. Link [Accessed: July 20, 2023].

8.Tan, Z., Wang, S., & Yang, Z. (2020). *Neural Machine Translation: A Review of Methods, Resources, and Tools*. *ScienceDirect*. Link [Accessed: July 2023].

9.Al-Rukban, A., & Saudagar, A. K. J. (2017). *Evaluation of English to Arabic Machine Translation Systems Using BLEU*. In Proceedings of the 2017 9th International Conference on Education Technology and Computers, pp. 228–232.

10.Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks*. Advances in Neural Information Processing Systems, Vol. 2.

11.Choudhary, H., Rao, S., & Rohilla, R. (2020). *Neural Machine Translation for Low-Resourced Indian Languages*. In Proceedings of LREC 2020. Link

12.Madaan, P., & Sadat, F. (2020). *Multilingual Neural Machine Translation involving Indian Languages*. In WILDRE5 Workshop. Link

13.Das, A., Yerra, P., Kumar, K., & Sarkar, S. (2016). *A Study of Attention-Based Neural Machine Translation Model on Indian Languages*. WSSANLP 2016. Link

14.Shah, P., & Bakrola, V. (2020). *Neural Machine Translation System of Indic Languages – An Attention-Based Approach*. Link

14.Srivastava, S., & Tiwari, R. (2019). *Self-Attention Based End-to-End Hindi-English Neural Machine Translation*. Link

16.Patel, R. N., Pimpale, P. B., & Sasikumar, M. (2017). *Machine Translation in Indian Languages: Challenges and Resolution*. Link

17.Ram, V. S., & Devi, S. L. (2023). *Hindi to Dravidian Language Neural Machine Translation Systems*. In RANLP 2023. Link

18.Kumar, A., & Singh, A. (2019). *Transformer-Based Neural Machine Translation for English to Hindi*. In Proceedings of the 12th International Conference on Natural Language Processing (ICON 2019), pp. 112-119. Link

19. Goyal, P., & Gupta, R. (2021). *Comparison of LSTM and Transformer Models for Hindi-English Neural Machine Translation*. In Proceedings of the 4th International Conference on Artificial Intelligence and Natural Language Processing (AINLP 2021), pp. 134-140. Link

20.Kumar, V., & Sharma, S. (2020). *Deep Learning Models for Machine Translation of Indian Languages: A Case Study on English to Hindi Translation*. In Proceedings of the 7th International Conference on Natural Language Processing (NLP 2020), pp. 101-107. Link