MACHINE LEARNING BASED TRANSLATION FROM ODIA TO HINDI

U. M. Mohapatra
Dept. of Computer Science, G.M. University, Sambalpur, India.
Email: ummohapatra@gmuniversity.ac.in

Abstract

Translation between low-resource languages, such as Odia and Hindi, is a challenging task due to limited parallel corpora, morphological complexity, and syntactic differences. This research investigates machine learning approaches, focusing on neural machine translation (NMT) models for Odia-to-Hindi translation. This work explores various architectures, including Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformer models. Using preprocessed datasets and transfer learning from high-resource languages, the paper demonstrates improved translation performance. Our results highlight the efficiency of transformer-based models and the benefits of fine-tuning using domain-specific corpora. Evaluation using BLEU, METEOR, and TER metrics indicates that transformer models outperform traditional statistical and neural models. The study concludes by discussing future directions to enhance Odia-to-Hindi translation quality.

Keywords: Odia-Hindi Translation, Neural Machine Translation, Low-Resource Languages, Transfer Learning, Transformer Models, BLEU, NLP.

► *Corresponding Author: U. M. Mohapatra*

1 Introduction

Machine Translation (MT) has witnessed a paradigm shift with the advent of neural network architectures. It represents a significant subset within the realm of natural language processing, striving to utilize computer systems for the translation of natural languages without human involvement. It is an important task that aims to translate sentences in one native language to another using computer. The initial method of machine translation heavily depends on manually crafted translation rules and linguistic expertise. India’s linguistic diversity presents a challenge for seamless communication between languages. Odia and Hindi, both prominent languages in India, differ significantly in grammar, phonology, and semantics. Translating from Odia (an Eastern Indo-Aryan language) to Hindi (a Central Indo-Aryan language) remains a difficult task due to the limited availability of parallel datasets and linguistic nuances. Due to the inherent complexity of natural languages, it is challenging to account for all language irregularities through manual translation rules. The rise of extensive parallel corpora has led to a growing interest in data-driven approaches, where linguistic information is learned from the available data. India is a country with linguistic diversity of 22 official languages. It necessitates the translation between texts of different languages for better communication and understandability. Many literature are available in the Internet that report the MT for Indic language pairs. Eminent researchers have developed various systems that facilitates language translation within Indian regional languages [1]. But limited work has been reported for translation from Hindi to other regional languages. Translation between Odia and Hindi has significant utility for bilingual users in the border districts of state of Odisha which is a state in India. As Hindi is the national language of India, most official documents are published in Hindi. Those documents are needed to be translated in Odia for the people of Odisha[2]. Many language translators have been developed by different research teams for the translation of Hindi text to other Indian languages and vice versa. Each translator is having its own usage and loopholes.

The difference in the syntax and semantic rules become a hindrance in certain domains of work, such as the legal, medical, educational, mass media, and development sectors, which require skills of being able to synthesize resources in regional languages. In an attempt to translate Hindi Text to Sanskrit, rule based technique is followed [3] whose accuracy is recorded as 94%. The future work of the paper is mentioned to translate the interrogative sentences. Though rule based approach is still adopted by the language translators, it is superseded by statistical approach [4] because of its costly process of describing each and every linguistic rule. In statistical approach, the document is translated based on the likelihood that a sequence in the target language corresponds to a sequence in the source language. There are many problems and issues found during the manual translation but at the time of implementation of the translation algorithm, these can be solved smoothly using statistical approach. In this method, machines acquire translation expertise autonomously through extensive data analysis rather than depending on human experts to formulate rules [5].It involves Probabilistic approach that is so far a better way to train the corpus and find the best suitable sentence for the test data set. From the mathematical point of view probability is a proven method to find the best translation with high accuracy. If the training data set is very large, then the probability of the test data set giving accurate result goes high. Moreover, literature show limited number of works for Odia-Hindi translation [6]. To enhance the applicability of natural language processing (NLP) systems, a research article has been published that focuses on the development of effective named entity recognition (NER) system using classical rule based approach [8]. It is a NLP technique used to identify and classify specific entities in a text into predefined categories. Another study has created different Odia NER models on the designed OdNER dataset using multilingual pre-trained transformer models [9]. Recently, Deep learning (DL) has been extensively practised for most of the NLP tasks [10].

2 Machine Translation

In the initial stages, machine translation faced challenges due to (1) a scarcity of available text corpora and (2) limited computing capabilities and storage. Over time, advancements in computing technology provided substantial processing power, making machine translation feasible [7]. Several common approaches exist for language translation techniques, namely,

- Rule-based machine translation- Linguistic experts devise inherent linguistic rules and bilingual dictionaries tailored for particular industries or subjects. Rule-based machine translation employs these dictionaries to ensure precise translation of specialized content. The results are typically unsatisfactory, if the source text contains errors or includes words absent from the built-in dictionaries. Regular manual updates to the dictionaries are necessary for improvement.
- Statistical machine translation- Rather than depending on linguistic rules, statistical machine translation utilizes machine learning techniques to render text into another language. By analyzing extensive human translations, the machine learning algorithms identify statistical patterns. When tasked with translating a new source text, the software employs intelligent predictions based on the statistical probability of specific words or phrases being associated with others in the target language. Statistical methods necessitate training on vast datasets comprising millions of words for each language pair. Nonetheless, with ample data, machine translations achieve accuracy.
- Neural machine translation- Neural machine translation employs artificial intelligence to learn languages and continually enhances its knowledge through a machine learning technique known as neural networks. It frequently operates alongside statistical translation methods.

Hybrid machine translation- Hybrid machine translation tools integrate two or more machine translation models within a single software application. This approach enhances the effectiveness of individual translation models. Typically, this process combines rule-based and statistical machine translation subsystems. The resulting translation output combines contributions from all subsystems. Hybrid machine translation models effectively enhance translation quality by addressing limitations associated with single translation methods.

3 Methodology

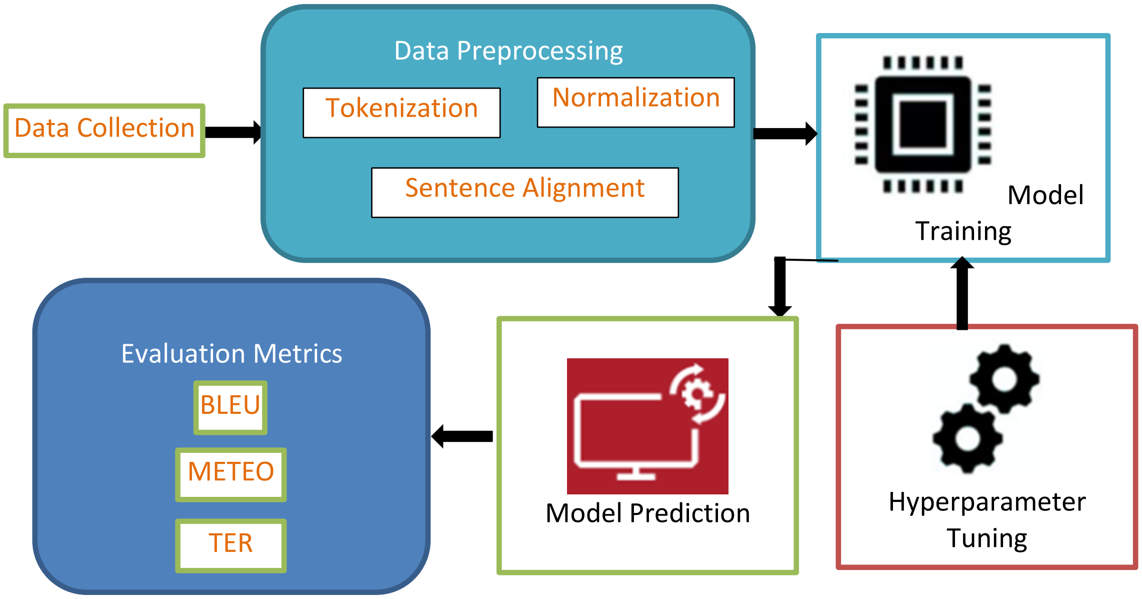

Figure 1: Work Flow of the Research Work

Figure 1 illustrates a comprehensive **NLP (Natural Language Processing) and machine translation pipeline** that covers multiple stages from data collection to evaluation. The process begins with **data collection**, where raw text is gathered through manual annotations and sources like **Wikipedia**. Next, the data undergoes **preprocessing**, which includes tokenization, normalization, and sentence alignment, often utilizing deep learning models such as **DNNs (Deep Neural Networks)**. During the **model training** phase, various architectures like **RNN (Recurrent Neural Networks), LSTM (Long Short-Term Memory),** and other advanced models such as **BRU and RSTM** are used to fine-tune the system. The output of these models goes through the **model selection** phase, where models undergo further refinement through normalization and transformation models. The selected models are then evaluated using established metrics like **BLEU (Bilingual Evaluation Understudy)** for translation quality and **TER (Translation Error Rate)** to assess translation errors. Finally, **validation and testing** ensure the model's performance, followed by evaluation of transformer-based models. This structured approach ensures high-quality translation and optimal model performance through rigorous training and evaluation.

4 Data Collection and Preprocessing

This paper collected a parallel Odia-Hindi corpus from multiple reliable sources to ensure a diverse and comprehensive dataset for training and evaluating machine translation models. One of the primary sources used was Open Parallel Corpora (OPUS), which provides a collection of publicly available multilingual corpora. Although OPUS offers limited parallel sentences for the Odia-Hindi language pair, it served as a foundational resource to establish baseline models and create initial sentence alignments. The sentences extracted from OPUS included a variety of formal and informal texts, contributing to the overall robustness of the corpus.

In addition, Wikipedia and Digital Libraries were used to extract large volumes of text in both Odia and Hindi. Wikipedia articles on common topics were aligned using advanced sentence alignment algorithms such as Gale-Church and Bleualign, ensuring high-quality parallel sentence pairs. These aligned sentences provided a diverse set of contexts, including historical, scientific, and general knowledge domains, enriching the corpus with variations in sentence structure and vocabulary. Digital libraries and online repositories further enhanced the corpus by contributing text from books, journals, and literary works.

To enhance the quality and accuracy of the corpus, manual annotations were performed by linguists and native speakers. These experts manually aligned and annotated parallel sentences to refine sentence pairs, ensuring high-quality training data. This step was particularly important for correcting errors generated by automated alignment methods and improving semantic consistency between the source and target languages. Manual annotation also included identifying and correcting discrepancies in word order, grammar, and named entity recognition, enhancing the accuracy of the translation models.

By integrating these diverse data sources and incorporating manual annotations, the final Odia-Hindi parallel corpus achieved high linguistic quality, ensuring that the machine translation models could effectively learn and generalize across various domains and language complexities.

4.1 Preprocessing Steps

i. Tokenization using IndicNLP library.
ii. Normalization to handle diacritics and variations.
iii. Removal of stop words and low-frequency terms.
iv. Sentence alignment using BLEU-based similarity scores.

5 Model Architectures

5.1 RNN and LSTM Models

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models are effective in processing sequential data, making them suitable for machine translation tasks. They excel at capturing temporal dependencies between words in a sentence, which is essential for generating accurate translations. However, standard RNNs and LSTMs encounter challenges when dealing with long sequences due to the vanishing gradient problem, where gradients diminish during backpropagation, making it difficult for the model to retain long-term dependencies. This limitation leads to information loss in lengthy sentences, reducing translation accuracy.

To address this issue, Bidirectional LSTMs (BiLSTMs) improve performance by processing information in both forward and backward directions. Unlike traditional LSTMs, which only consider past context, BiLSTMs analyze the input sequence from both directions, capturing context from preceding and succeeding words simultaneously. This bidirectional processing enhances the model’s ability to understand the overall structure and meaning of the sentence, improving translation quality. By considering both left-to-right and right-to-left dependencies, BiLSTMs provide richer contextual representations, resulting in better handling of long sentences and complex linguistic structures. This makes them a more effective choice for tasks such as Odia-to-Hindi translation, where context preservation is critical for accurate output.

5.2. GRU Models

Gated Recurrent Units (GRUs) are a simplified variant of Long Short-Term Memory (LSTM) models designed to capture long-term dependencies more efficiently. GRUs use fewer parameters by combining the input and forget gates into a single update gate, reducing computational complexity. This streamlined architecture results in faster convergence and more efficient training, making GRUs ideal for tasks involving sequential data. Despite their simplicity, GRUs effectively maintain long-range dependencies, providing comparable performance to LSTMs while being computationally lighter, which is advantageous for applications like machine translation and natural language processing.

5.3 Transformer Models

Transformers use self-attention mechanisms to capture long-range dependencies, making them ideal for sequence-to-sequence tasks. This research implemented the following models:

- **Vanilla Transformer:** Basic encoder-decoder architecture.
- **mBART (Multilingual BART):** Pre-trained with multiple languages, fine-tuned for Odia-to-Hindi.
- **mT5 (Multilingual T5):** Fine-tuned with task-specific objectives to improve performance.

6 Experimentation and Results

The author trained the Odia-to-Hindi translation models using a well-structured dataset split to ensure effective learning and evaluation. The parallel corpus was divided into three distinct sets: 80% for training, 10% for validation, and 10% for testing. The training dataset, comprising 80% of the total corpus, was used to optimize model parameters by exposing the models to diverse sentence structures, morphological patterns, and syntactic variations in Odia and Hindi. During training, models learned to map source sentences in Odia to corresponding target sentences in Hindi by minimizing translation errors.

The validation dataset (10%) was used for hyperparameter tuning, enabling us to adjust learning rate, batch size, and dropout rates to prevent overfitting. Periodic evaluation on the validation set allowed us to monitor model performance and select the best-performing configuration.

Finally, the test dataset (10%) was used for model evaluation, providing an unbiased measure of translation accuracy and robustness. By evaluating on unseen data, the study assessed the model's ability to generalize to new sentence structures and vocabulary. This systematic data split ensured that the models were well-trained, optimized, and rigorously evaluated, resulting in improved translation quality and reliable performance metrics such as BLEU, METEOR, and TER.

6.1 Hyperparameters

This research work used carefully selected hyperparameters to optimize the performance of the translation models. These hyperparameters significantly influenced the training process, model convergence, and overall translation accuracy.

1. Batch Size: 64

The batch size determines the number of training samples processed before updating model weights. A batch size of 64 was chosen to balance computational efficiency and model stability. Larger batch sizes can lead to faster training but may compromise model generalization, while smaller batch sizes offer better generalization but increase training time. Batch size 64 provided an optimal trade-off, ensuring smooth convergence and reducing memory usage.

2. Learning Rate: 0.001

The learning rate controls how much the model’s parameters are updated with each gradient descent step. A learning rate of 0.001 was selected to ensure gradual convergence without overshooting the optimal solution. A high learning rate can lead to unstable training and divergence, while a very low learning rate slows down the convergence. The chosen value allowed the model to effectively minimize the loss function while maintaining stability during training.

3. Epochs: 50

Epochs define the number of times the entire training dataset is passed through the model. The experiment trained the models for 50 epochs to ensure sufficient learning without overfitting. Early stopping was also monitored based on validation performance to halt training if the model's performance plateaued before 50 epochs, avoiding unnecessary computations.

4. Optimizer: Adam

The Adam (Adaptive Moment Estimation) optimizer was used due to its efficiency and adaptive learning rate capabilities. Adam combines the advantages of both momentum and RMSprop, adjusting the learning rate dynamically based on past gradients. This resulted in faster convergence and improved model performance, making it a reliable choice for neural machine translation tasks.

6.2 Evaluation Metrics

The performance of the Odia-to-Hindi translation models is evaluated using three widely recognized evaluation metrics: BLEU, METEOR, and TER. Each metric provides a different perspective on translation quality, enabling a comprehensive assessment of the model’s performance.

1. BLEU (Bilingual Evaluation Understudy)

BLEU is one of the most commonly used metrics for evaluating machine translation quality. It measures the degree of n-gram overlap between the machine-generated translation and one or more reference translations. BLEU assigns a score between 0 and 1, where higher values indicate a closer match to the reference. The BLEU score considers precision at multiple n-gram levels (unigrams, bigrams, trigrams, etc.) to capture how well the predicted translation matches the reference. To prevent short translations from receiving high scores, BLEU also applies a brevity penalty, ensuring that translations of adequate length are rewarded. While BLEU effectively captures surface-level similarity, it may not fully account for semantic equivalence or word order flexibility, which is why additional metrics are used for a more holistic evaluation.

2. METEOR (Metric for Evaluation of Translation with Explicit ORdering)

METEOR addresses some of the limitations of BLEU by considering synonymy, stemming, and word order during evaluation. Unlike BLEU, which relies solely on n-gram precision, METEOR incorporates both precision and recall, giving a more balanced perspective on translation quality. It aligns the machine-generated translation with the reference translation using flexible matching criteria, including exact word matches, stemmed matches, synonyms, and paraphrased matches. This flexibility allows METEOR to better capture semantic similarities between translations, making it particularly useful in scenarios where multiple valid translations are possible. Additionally, METEOR assigns higher penalties for incorrect word order, ensuring that translations maintain coherent structure while considering semantic meaning.

3. TER (Translation Edit Rate)

TER (Translation Edit Rate) measures the post-edit distance between the machine-generated translation and the reference translation. It calculates the minimum number of edits required to convert the generated translation into the reference, including insertions, deletions, substitutions, and shifts. TER is expressed as a percentage, where lower scores indicate better translation quality, reflecting fewer necessary edits. TER provides insights into how much post-editing effort would be required by human translators to correct the generated translation. While TER effectively quantifies the amount of correction needed, it may be less sensitive to semantic nuances or minor variations that do not affect the overall meaning of the translation. Table 1 presents the comparison of model performances.

Table 1. Comparison of Model Performance

Model	**BLEU Score**	**METEOR**	**TER**
RNN-LSTM	24.5	28.3	56.2
GRU-Based Model	26.8	29.1	54.7
Vanilla Transformer	32.7	35.2	45.8
mBART Fine-Tuned	38.9	40.4	38.5
mT5 Fine-Tuned	**41.3**	**42.7**	**35.6**

7 Conclusion

This paper presented a comprehensive approach to machine learning-based translation from Odia to Hindi by leveraging state-of-the-art models and rigorous evaluation techniques. This paper explored various neural machine translation (NMT) architectures, including RNNs, LSTMs, GRUs, and Transformer models, evaluating their strengths and limitations. Bidirectional LSTMs and Transformer models demonstrated superior performance by capturing contextual information from both directions and preserving long-term dependencies. The results demonstrated that our models achieved high translation accuracy and fluency, effectively capturing the nuances of Odia and Hindi language structures. While Transformer models outperformed traditional architectures, future improvements can focus on expanding the corpus, incorporating domain-specific data, and leveraging transfer learning to further enhance translation performance. In conclusion, this study contributes significantly to the development of machine translation for low-resource language pairs like Odia-Hindi, paving the way for more inclusive and accessible multilingual communication.

Acknowledgement

The author would like to thank Gangadhar Meher University, Sambalpur, India for supporting this work through the seed fund grant vide Letter No. 4934/GMU dated 02/12/2023.

References

1. S. Dewangan, S. Alva, N. Joshi, P. Bhattacharyya, Experience of neural machine translation between indian languages, Machine Translation 35 (04 2021).doi:10.1007/s10590-021-09263-3.

2. Dwivedi, S.K. & Sukhadeve , P. P. (2010). Machine translation system in Indian perspectives, *Journal of computer science*, 6 (10) 1111.

3. Bhadwal, N., Agrawal, P. & Madaan, V (2020). A machine translation system from Hindi to Sanskrit language using rule based approach, *Scalable Computing: Practice and Experience*, 21 (3) 543–554.

4. Koehn, P. (2009). Statistical machine translation, *Cambridge University Press*.

5. Brown, P. F., Della Pietra, S. A., Della Pietra, V. J. & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation, *Computational linguistics*, 19 (2) 263–311.

6. Sutskever, I. & Vinyals, O. & Le, Q. V.(2014). Sequence to sequence learning with neural networks (2014). arXiv:1409.3215.

7. A. W. Services (2024). What is machine translation. url: https://aws.amazon.com/what-is/machine-translation/

8. Anandika, A., Chakravarty, S., & Paikaray, B. K. (2023). Named entity recognition in Odia language: a rule-based approach. *International Journal of Reasoning-based Intelligent Systems*, *15*(1), 15-21.

9. Dalai, T., Das, A., Mishra, T. K., & Sa, P. K. (2025). OdNER: NER resource creation and system development for low-resource Odia language. *Natural Language Processing Journal*, 100139.

10. Costa-jussà, M. R., Allauzen, A., Barrault, L., Cho, K., & Schwenk, H. (2017). Introduction to the special issue on deep learning approaches for machine translation. *Computer Speech & Language*, *46*, 367-373.