

Self-attention encoder-decoder with model adaptation for transliteration and translation tasks in regional language

Shanthala Nagaraja, Kiran Y. Chandappa

Department of Information Science and Engineering, Global Academy of Technology, Bangalore, India

Article Info

Article history:

Received Oct 3, 2023

Revised Jul 10, 2024

Accepted Aug 12, 2024

Keywords:

Auto-encoder

Natural language processing

Neural network

Self-attention encoder-decoder
with model adaptation

Self-attention mechanism

ABSTRACT

The recent advancements in natural language processing (NLP) have highlighted the significance of integrating machine transliteration with translation for enhanced language services, particularly in the context of regional languages. This paper introduces a novel neural network architecture that leverages a self-attention mechanism to create an auto-encoder without the need for iterative or convolutional processes. The self-attention mechanism operates on projection matrices, feature matrices, and target queries, utilizing the Softmax function for optimization. The introduction of the self-attention encoder-decoder with model adaptation (SAEDM) represents a breakthrough, marking a substantial enhancement in transliteration and translation accuracy over previous methodologies. This innovative approach employs both student and teacher models, with the student model's loss calculated through the probabilities and prediction labels via the negative log entropy function. The proposed architecture is distinctively designed at the character level, incorporating a word-to-word embedding framework, a beam search algorithm for sentence generation, and a binary classifier within the encoder-decoder structure to ensure the uniqueness of the content. The effectiveness of the proposed model is validated through comprehensive evaluations using transliteration and translation datasets in Kannada and Hindi languages, demonstrating its superior performance compared to existing models.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Shanthala Nagaraja

Department of Information Science and Engineering, Global Academy of Technology

Rajarajeshwarinagar, (off Mysore Road), Ideal Homes Township, Bangalore-560098, Karnataka, India

Email: shanthala_12@rediffmail.com

1. INTRODUCTION

In today's modern world, effective cross-language communication and information access have become indispensable. The rapid expansion of the internet has led to a vast amount of digital information being generated in various languages, including Hindi and English [1]. Cross-language information retrieval (CLIR) enables users to find relevant data in different languages, making it particularly crucial for languages like Hindi and English, which have distinct linguistic frameworks and writing systems [2]. This study investigates the performance and effectiveness of CLIR techniques, specifically focusing on Hindi-to-English transliteration and translation [3]–[5]. Hindi holds significant importance as it is spoken by over 40% of Indians and acts as a lingua franca, bringing together people from diverse linguistic and geographical backgrounds [6]. Besides English, Hindi is one of the 22 officially recognized languages in India [7], [8]. Due to its official status, a considerable amount of official paperwork, legal documents, and communication are produced in Hindi, emphasizing the need for efficient and accurate CLIR systems to enhance accessibility to such critical information [9], [10]. Given its widespread usage, official status, and the growing digital

content, Hindi plays a crucial role in CLIR in India, facilitating intercultural communication and providing access to official information [11], [12].

For Hindi-to-English CLIR, transliteration is particularly useful, as it converts text between the Devanagari script (used in Hindi) and the Latin script (used in English) while preserving phonetic similarity [13], [14]. However, transliteration may not always capture the exact semantic meaning of the text. On the other hand, translation involves conveying the meaning of a text from one language to another, making it more suitable for transferring semantic information between languages like Hindi and English. Although translation-based techniques have shown advantages over transliteration algorithms in previous CLIR research, it remains uncertain if these findings apply to a wide range of language pairs and domains, especially for Hindi-to-English CLIR [15].

Researchers and linguists have explored various challenges related to transliteration and translation models. Translation studies have examined aspects such as cultural background, translation accuracy, and the impact of translation on literary works. On the other hand, most of the focus on transliteration studies has centered around developing tools and algorithms for text translation across scripts. However, the relative effects of transliteration and translation on cross-linguistic communication in the context of regional languages like Kannada, Hindi and their interaction with English have not been extensively investigated [16]. Recent advancements in deep learning and natural language processing (NLP) have significantly improved machine translation and transliteration for regional languages like Kannada. These developments have focused on overcoming resource constraints and language complexity while enhancing the accuracy, fluency, and effectiveness of language models. Neural machine translation (NMT) models utilizing deep learning techniques have proven to be more precise and reliable than traditional statistical and rule-based approaches [17], [18]. Furthermore, large-scale pre-trained language models like bidirectional encoder representations from transformers (BERT) and generative pre-trained transformers (GPT), along with their multilingual versions, have demonstrated remarkable performance in various NLP tasks, including translation and transliteration when trained on Hindi and Kannada data.

In conclusion, effective cross-language communication and access to information are essential in the digital era. The study aims to investigate the performance and effectiveness of Hindi-to-English transliteration and translation techniques in CLIR. Considering the significance of Hindi in India, accurate CLIR systems are crucial for fostering intercultural communication and breaking down language barriers in the country. Recent advancements in deep learning and NLP have shown promising results in enhancing translation and transliteration models, providing better access to information across different languages [19].

In this novel approach, two pre-trained BERT models from different domains are seamlessly integrated into a sequence-to-sequence model using adapter modules [20]. These adapters are introduced between BERT layers and fine-tuned, while latent variables during fine-tuning determine which layers utilize the adapters. This intelligent adaptation significantly enhances parameter efficiency and decoding speed. Testing the proposed framework against various NMT challenges demonstrates its effectiveness.

The paper introduces the iterative and length-adjustable non-autoregressive decoder (ILAND) [21], a unique paradigm for machine translation. ILAND employs a length-adjustable non-autoregressive decoder that uses a hidden language model to prevent low-confidence token creations while maximizing target sentence length. Comprising three sub-modules-token masker, length modulator, and token generator-ILAND collaboratively achieves its objectives. The token masker and token generator handle the masked language model, while the length modulator optimizes sentence length. The sequence-to-sequence training of the translation model is effectively demonstrated. The concurrent training of the length modulator and token generator, which share similar structures, contributes to the model's superior performance compared to other non-autoregressive decoders, providing empirical validation.

Another proposed technique [22] introduces knowledge-aware NMT, incorporating additional language properties at the word level using recurrent neural networks (RNN). The sentence level uses an RNN encoder to encode these word-level feature units. Additionally, a knowledge gate and an attention gate control the quantity of information from various sources to assist in decoding and constructing target words. This approach proves efficient in enhancing NMT performance by leveraging prior translation knowledge from the source side of NMT's training pipeline. It enables NMT to effectively incorporate past translation information and cross-language translation data, resulting in improved translation accuracy and quality.

To enhance NMT model performance for constrained resources and language pairs, a novel strategy is proposed for standardizing NMT model training [6]. This approach involves training the model to predict target training texts using word and sentence embeddings as well as categorical outputs (i.e., word sequences). By pre-training word and phrase embeddings on substantial monolingual data corpora, the model gains the ability to generalize beyond the translation training set, improving translation accuracy.

A unique improvement to generative adversarial networks-neural machine translation (GAN-NMT) is suggested by incorporating deep reinforcement learning-based attention optimization into the generator and

a convolutional neural network into the discriminator [23]. This enhancement addresses the challenge of unusual terms in low resource languages (LRLs) and improves the assimilation of source sentence representations. Additionally, a novel joint embedding of subwords and sub-phonetic representations of sentences is utilized as GAN input, enabling the model to learn superior representations and generate context vectors more suitable for LRLs than traditional techniques.

A multi-task multi-stage transitional (MMT) training framework is proposed [24], utilizing a bilingual conversation translation dataset and extra monolingual conversations. This framework involves three steps: sentence-level pre-training on a sizable parallel corpus, intermediate training with additional monolingual conversations and unique tasks (utterance and speaker detection) to model conversation coherence and speaker characteristics, and context-aware fine-tuning with a gradual transition. This incremental transition strategy smoothly switches monolingual conversations to multilingual ones, facilitating a more refined training process.

To aid NMT systems, a straightforward and useful model of the potential cost of each target word is introduced [25]. This model learns a representation of future costs based on the previously created target term and its context, which assists in NMT model training. During decoding, the learned representation of future costs in the current time phase is utilized to produce the next target word, enhancing translation accuracy.

Research focuses on deep learning-based Hindi-to-English transliteration and translation in CLIR. Objective: improve information access and cross-cultural communication. Develop accurate, efficient CLIR systems using NLP advancements. Benefit industries (business, healthcare, and education) needing multilingual information. Investigate deep learning model adaptability and domain-specific effectiveness. Enhance intercultural understanding and aid travel, hospitality, and commercial sectors. Translation and transliteration enable market expansion and economic growth in Hindi and Kannada languages. Overall, research aims to break language barriers, foster cross-lingual collaboration, and facilitate efficient information retrieval.

Novel self-attention encoder-decoder with model adaptation (SAEDM): the proposed introduces a neural network architecture that utilizes self-attention to construct an encoder-decoder without the need for iterations or convolutional operations. This novel approach enhances the efficiency and effectiveness of the encoder-decoder, allowing for more accurate and faster processing of word and sequence embeddings.

Improved transliteration and translation performance: the study demonstrates the superiority of the proposed model over existing systems in transliteration and translation tasks for Kannada and Hindi languages. By incorporating lexical weighting and fine-tuning techniques, the proposed system achieves substantial improvements in accuracy, showcasing its potential in enhancing cross-lingual information retrieval and communication.

Student-teacher model for model compression and domain adaptation: the research introduces a student-teacher model approach for model compression and domain adaptation. This technique facilitates effective knowledge transfer between models, enabling the system to adapt to different domains and languages. The use of student-teacher models enhances the model's ability to perform in various linguistic contexts, making it a versatile and adaptable solution.

Integration of binary classifier for content originality: the incorporation of a binary classifier with the auto-encoder ensures content originality and mitigates the risk of plagiarism. This contribution adds an extra layer of credibility and reliability to the model, making it suitable for applications where content authenticity is essential, such as in academic research, legal documents, and official communications.

The research work in this paper is organized as follows: in the section 1, a brief introduction is given about the challenges across text-pre-processing, how the transliteration and translation models have been built that overcomes the challenges in various languages, and the breakthroughs involved in processing the Hindi and Kannada language. In section 2 a neural network without iterations or convolutional operations is developed that focuses on a self-attention mechanism to build an encoder-decoder model. In section 3 the dataset details and results for transliteration and translation models are shown. The last section is conclusion.

2. PROPOSED METHOD

The provided information describes a neural network architecture that employs self-attention mechanisms and domain adaptation techniques. The network takes word or sequence embeddings as input and consists of multiple embedded layers with self-attention and multi-layer perceptron (MLP) components. It utilizes recursive transfer and auto-encoder with masked layers to handle complex data and overcome gradient vanishing. Model compression is achieved using student-teacher models, while domain adaptation ensures good performance across different domains. Figure 1 shows the proposed workflow. The system design involves character-level word-to-word embedding architecture with attention mechanisms. Beam search and a binary classifier are utilized for NLP tasks and content originality assurance, respectively. This comprehensive architecture aims to process and adapt to various domains efficiently.

Figure 1 illustrates a machine learning pipeline for processing natural language data, starting with the input of word sequence embeddings-vector representations of text. An encoder component processes these embeddings to condense the information into a more manageable form for the model to handle. Following this, a decoder takes the encoded data and constructs an output sequence, which can be text, a summary, translation, or another form depending on the application. The model adaptation stage suggests that the base model can be fine-tuned to better fit specific tasks or new data domains, enhancing its performance on tasks different from the ones it was initially trained on. Finally, in-model adaptation and out-model adaptation imply that fine-tuning can occur within the existing model structure or through external modifications to the architecture, respectively.

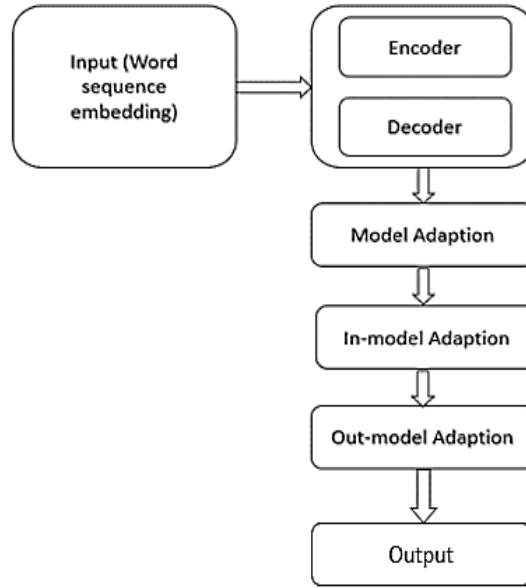


Figure 1. Proposed workflow

The above network is a unique type of neural network that does not involve iterations or convolutional operations. Instead, it relies on a self-attention mechanism to construct an auto-encoder. The input provided to the network can be either a word embedding or a sequence embedding. The encoder decoder consists of A embedded layers, multiple self-attention layers, convolutions, and masked multiple self-attention layers for mop . The self-attention layers along with the text is neither created nor masked. An input is provided as x^p with the sequence, embedding the transformation via a weight matrix subjected to L^p , the projection matrix denoted as T^p with the feature matrix shown by N^p , self-attention mechanism is later applied on L^p, T^p, N^p irrespective, Softmax function denoted by \mathcal{H} .

$$\alpha(L^p, T^p, N^p) = \text{seg}(L_{p+1}^{p+1} \dots \dots L_X^{p+1})R^p \quad (1)$$

$$L_x^{p+1} = \mathcal{H}L_x^p L_x^Z (I_m)^{-2} (N_x^p) \quad (2)$$

$$L_x^p, T_x^p, N_x^p = L^p R_x^l, T^p R_x^t, N^p R_x^n, \quad (3)$$

Henceforth L_x^p, T_x^p, N_x^p represents the x -th query with the feature matrix for the p -th layer. $\{R_x^l, R_x^t, R_x^n\} \in S^{m_d}$ depicts the variable matrix m_d representing the dimension for the model irrespectively. A MLP is a fully connected network with the activation function that is applied on each function.

$$L^{n+1} = \beta(\alpha(L^p, T^p, N^p)) + L^p \quad (4)$$

However, L^{n+1} is the preliminary stage with relevant feature information however L^n is added to develop the connections that overcomes gradient vanishing. This process sequencing is depicted as I_α , denoted by the source as M^{r+1} . The α utilises different layers to learn from the source.

$$M^{r+1} = I_\alpha^{p+1}(L^p, T^p, N^p) \quad (5)$$

The α utilises different layers to learn from the source. Henceforth this $[...]_z (z \in \{1, 2, \dots, Z\})$ represents the Z similar layers stacked up with each other. The outcome L_z for the Z -th attention layer denoted by the final outcome transmitted. The transfer of the auto-encoder by a translation model which predicts the target. The difference between the auto-encoder by the masked layer that represents the output designed dynamically. The output is decoded which estimates the probability of $\log e(y_m | y < m, \mathcal{H})$.

$$[L^z = i_\alpha^z(L^{z-1}, T^{z-1}, N^{z-1})]_z \quad (6)$$

2.1. Entropy

The proposed approach here become accustomed by fine-tuning the model and out-model training to concentrate on training and perform the model compression. This includes two-sub sections known as the student and teacher model. The student loss is determined by two components.

- The loss associated with the probability and prediction of the label by incorporating negative log entropy function.

$$\text{Entropy}(\delta_d; F) = \sum_{(c,d \in F)} \sum_{m=1}^w -U(y_m) * \log e(y_m | y < m, \delta_d) \quad (7)$$

- The model compression computes the loss in between the output probability and the student teacher model.

$$\text{Entropy}_m(\delta_d; F, \delta_d) = \sum_{(c,d \in F)} \sum_{m=1}^w -l(y_m | y < m, \delta_d) * \log e(y_m | y < m, \delta_d) \quad (8)$$

The $n(y_m | y < m, \delta_d^{\wedge})$ with δ_d represents the parameter for the student model. $\delta_d^{\wedge} = \delta_d *$, the method to average $\delta_d = \frac{1}{z} \sum_z \delta_d^{(x)}$, the weight approach is represented as $\delta_d(z) = \sum_z g - p(S \delta_d^{(z)}) \delta_d^{(z)}$, wherein $[g-p]$ represents the normalized function for z -th evaluation for the parameter $\delta_d^{(z)}$ that represents an ensemble model. The averaging and weighted-averaging approach in the student model is applied to gain necessary information by accumulation of information by the iterations of the teacher model.

2.2. Model adaptation

The in-model and out-model compute the pre-training of the model. In each level the model value is the added advantage for the former iteration for the purpose of in-model parameters. The process is iterated to achieve mutual transfer of data. The in-model and out-model features are transmitted across each other at model level to transmit data henceforth ensuring good performance. The model efficiency is evaluated across source and target domain data F_d, F_h is partitioned into the training sets F_d^n, F_h^n and model and the evaluation pair consist of $F_d^{\text{com}}, F_h^{\text{com}}$ that is further responsible to train and evaluate the model as shown in Algorithm 1.

Algorithm 1. Consists of the model adaptation algorithm for the proposed model that consists of two stages In the initial stage, the main aim is to complete the initialization for the in-model and out-model parameters.

- The μ function is responsible to train the model for the objective function F_d^n and the parameter used along with this $\delta_h^{(p+1)}$, initialized by $\text{Entropy}_m(\delta_d; F_d^n)$ retained for the source.
- In this step the recursive transfer of data is established in between in-model and out-model adaptation.
- The p function transfers the model, its main goal is to utilize this with self-knowledge function loss denoted as $\text{Entropy}(\delta_h^{(t-1)}; F_d^n)$ and the $\text{Entropy}_m(\delta_d^{(t-1)}; F_h^n)$, used along the training sets F_h^n .
 - The model transfer for in-domain parameter set δ_d is initialized through the previous out-domain model parameter as $\delta_h^{(t-1)}$. The initialization is carried out while optimizing the model is performed through the in-model and repeated for source domain.
 - The β model is utilised for the evaluation purpose via the ensemble activation function used for evaluation purpose for ensuring the performance of $\delta_d^{(t)}$ for building a set of F_d^{com} by the ensemble parameter depicted as δ_d .

Input: Training of $\{F_d^n, F_h^n\}$, represents the lists as, $\{F_d^{com}, F_h^{com}\}$, with level T.

Step 1: train in – model

Step 2: $\delta_d^{(p+1)} \leftarrow \text{tr}(\text{Entropy}(\delta_d; F_d^n))$

Step 3: train out – model

Step 4: $\delta_h^{(p+1)} \leftarrow \text{Entropy}(\delta_h; F_h^n)$

Step 5: Initialize in – model and out – model ensemble model parameters

Step 6: $\delta_d \leftarrow \delta_d^{(p+1)}, \delta_h \leftarrow \delta_h^{(p+1)}$

Step 7: for $t = 1, 2, \dots, T$ do

Step 8: in – model training=Transfer training model and computation

Step 9: $\delta_d^{(t)} \leftarrow \vartheta \text{Entropy}(\delta_h^{(t-1)}; F_d^n) \text{Entropy}_m(\delta_h^{(t-1)}; D_b^1 \tau_b))$

Step 10: $\delta_d \leftarrow \beta(F_d^{com}, \delta_d^{(t)})$

Step 11: train out – model=Transfer training model and computation

Step 12: $\delta_h^{(t)} \leftarrow (\text{Loss}(\delta_h^{(t-1)}; F_h^n) \text{Entropy}_m(\delta_d^{(t-1)}; F_h^n \delta_h))$

Step 13: $\delta_h \leftarrow \beta(F_h^{com}, \delta_h^{(t)})$

Step 14: end for

Output: in model training δ_d ; out model training δ_h

2.3. System design

Neural networks based on words can offer an end-to-end solution to the complexities associated with the huge number of words. However, character-level methods can also be used to evaluate the complexity associated with noise, alterations, and errors. These methods use word-to-word model embeddings to assess the complexity of text.

2.3.1. Pre-processing and post-processing

Input pre-processing: when an input word is uttered, it is normalized by converting all letters to lowercase, removing repeated characters, transforming diacritics into standard 7-bit American standard code for information interchange (ASCII), and converting emojis and emoticons involving punctuation into hashtags. During training, foreign words are tagged as hashtags and the output is aligned with the input through these hashtags. This ensures that the model learns to identify foreign words and transfer them into hashtags that are identical to the input. Additionally, emojis, emoticons, and punctuations are converted into hashtags during training and prediction. After training, a post-processing step converts the hashtags back to words in the source. If the input and output are aligned, this step is performed before removing the tokens [+] and [-]. However, in the final output, the words along with the [+] token are merged and the [-] tokens are replaced with a white space that splits a word into multiple words.

2.3.2. System architecture

A character-level word-to-word embedding architecture is a type of neural network that uses character-level embedding to represent words. This type of architecture is often used for tasks such as machine translation and text classification. The model $J(c|d)$ that generates an input d for target c . The proposed model consists of an attention mechanism which consist of gated recurrent unit (GRU). The initial stage in the proposed model which consists of recurrent neural networks and non-recurrent connections essential for training purpose. The Softmax layer for the proposed model's output to the final sequence output c . The entropy function evaluated for loss for time average over c_x . Beam search is a technique used in NLP to find the most likely sequence of words in a sentence. It works by keeping track of a fixed number of candidate sequences, and then at each step, predicting the next word in the sequence with the highest log-likelihood. The beam size is a hyper-parameter that controls how many candidate sequences are kept track of.

2.3.3. Binary classifier

As mentioned earlier, an autoencoder is enhanced with a binary classifier by integrating an attention mechanism. This attention mechanism is employed to create a task-specific contextual representation. The purpose of this augmentation is to avoid plagiarism and ensure the originality of the content $I_x(h)$ of (h) .

$$I_x(h) = \sum_{v=1}^g \vartheta_v u_v$$

$$\vartheta_v = \frac{\exp(u_v)}{\sum_{v'}^g \exp(u_{v'})} \quad (9)$$

$$u_v = (j_s)^y \tanh(N_s i_v)$$

Here j_s and N_s are the related constraints. Hence a binary classifier for $I_x(h)$ is shown as in the proposed model this classifier is trained to maximize the entropy. In this context, the weights are used to determine the importance or relevance of various target words within the sentence. To achieve this, the classifier is employed to calculate attentional weights for the target words during the model training process.

$$\beta_{ie}^i(h; \gamma_{jk}^j) = \log(d|h; \gamma_{jk}^j) \quad (10)$$

$$\beta_{ps}(k, h; \gamma_{ps}) = \sum_{v=1}^g (1 + P_v) \log(h_v | v, h < v; \gamma_{ps}) \quad (11)$$

Here P_v depicts the attention weight associated along the h_v to obtain (11) and γ_{ps} represents the variable related to ps. The model development modifies the magnitude of the updated parameter while preserving its direction. This ensures that task-specific words and task-shared words are consistently updated during training. The primary benefit of incorporating lexical weighting is the ability to train at the word level rather than the sentence level. This approach is advantageous as it reduces the time required for training the model while maintaining effectiveness.

3. RESULT EVALUATION

This section presents the analysis of results obtained using the SAEDM model for translation and transliteration. The model's performance is evaluated based on accuracy, and a comparative study is conducted with other transliteration methods. The main objective is to improve information transfer in Kannada and Hindi languages by enhancing the model's effectiveness. Details of the dataset used for transliteration and translation are provided. Simulations are conducted on an INTEL Core i7 processor, utilizing Python and deep learning libraries, with 8 GB random access memory (RAM) and a 64-bit Windows OS. The study aims to demonstrate the proposed model's efficiency in achieving better transliteration and translation results.

3.1. Dataset details

The Dakshina dataset [26] is a comprehensive and diverse collection of Indian languages, designed to support NLP research in Indian languages. It includes text from various domains and sources, encompassing Hindi and Kannada languages, among others. The dataset is aimed at facilitating NLP studies for Indian languages. Researchers can access text from different genres, such as news stories, social media content, and official documents. The availability of text in both its original script and transliterated form ensures accessibility even for those unfamiliar with the script. Moreover, the dataset provides annotations for named entities, part-of-speech identifiers, and sentence boundaries, making it valuable for information extraction and machine translation tasks. Overall, the Dakshina dataset serves as a valuable resource for academic researchers exploring Indian languages like Hindi and Kannada and their applications in NLP.

The web and translation (WAT) collection incorporates parallel corpora for multiple Indian languages, including Hindi and Kannada. The most recent version, WAT2021 in research [27], was released in 2021 and is part of the workshop on Asian translation (WAT) joint initiative. This dataset includes multilingual translation data and corresponding details in English. The compilation consists of approximately 1.5 million phrase pairs extracted from diverse sources such as news, communication, information technology, legal, and scientific content. Each sentence in the dataset has been tokenized, normalized, and aligned at the sentence level, ensuring its usability for various language-processing tasks.

3.1.1. Transliteration

The evaluation of transliteration models utilizes publicly available transliteration corpora, with a major portion of the data obtained from the Dakshina corpus [26]. Figure 2 shows the accuracy value for transliteration in Kannada language. Figure 3 shows the accuracy value for transliteration in Hindi language. The obtained results are then compared between the existing system and the proposed system for the Hindi and Kannada language, using data from the Dakshina corpus. The result evaluated is represented in the following which shows that the accuracy for the existing system [27], [28] is 77.18 and the PostScript (PS) value generates a value of 85.87% for Kannada language and for Hindi language the existing system [27] generates a value of 60.56 and the PS generates a value of 85.56%.

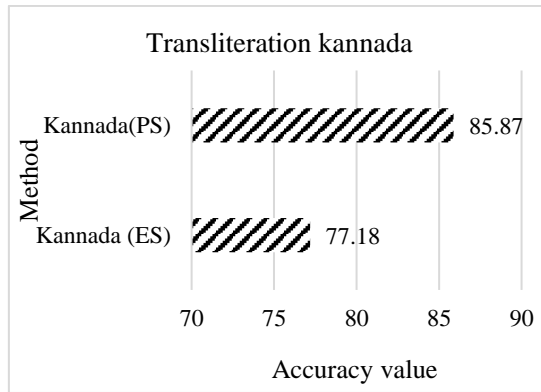


Figure 2. Accuracy value for transliteration in Kannada language

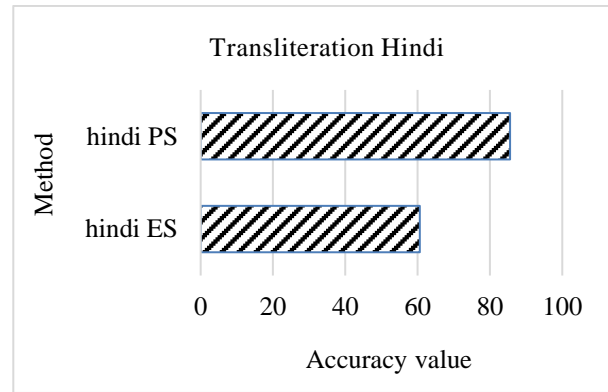


Figure 3. Accuracy value for transliteration in Hindi language

3.1.2. Translation

The evaluation of models in this study is conducted using bilingual evaluation understudy (BLEU) scores, and the assessment annotations include SacreBLEU signatures for both Indic-English21 and English-Indic22 translations, ensuring consistency, and reproducibility across models. Figure 4 shows the accuracy comparison for translation in Kannada language. Figure 5 shows the accuracy comparison for translation in Hindi language. The publicly available translation corpora are compiled, with a significant portion of the data sourced from WAT2021 [27]. The results obtained are then compared between the existing system and the proposed system for the Kannada language from WAT2021. mBART in research [29] generates an accuracy value of 4.3, Google (GOOG) in research [30] generates an accuracy value of 25.9, Microsoft (MSFT) in research [31] generates a value of 25.4, TF in research [32] generates a value of 26.8, MT5 in research [33] generates a value of 28.5 whereas the existing system generates a value of 36.2 and the proposed National Native Title Tribunal-PostScript (NNTT-PS) generates a value of 56.69 for Kannada language. The OPUS in research [29] method generates a value of 13.3, mBART in research [29] generates an accuracy value of 33.1, GOOG [30] generates an accuracy value of 36.7. MSFT [31] generates a value of 38, TF [32] generates a value of 38.8, MT5 [33] generates a value of 39.2 whereas the existing system generates a value of 40.3 [32] and the proposed NNTT-PS generates a value of 77.5497% for Hindi language.

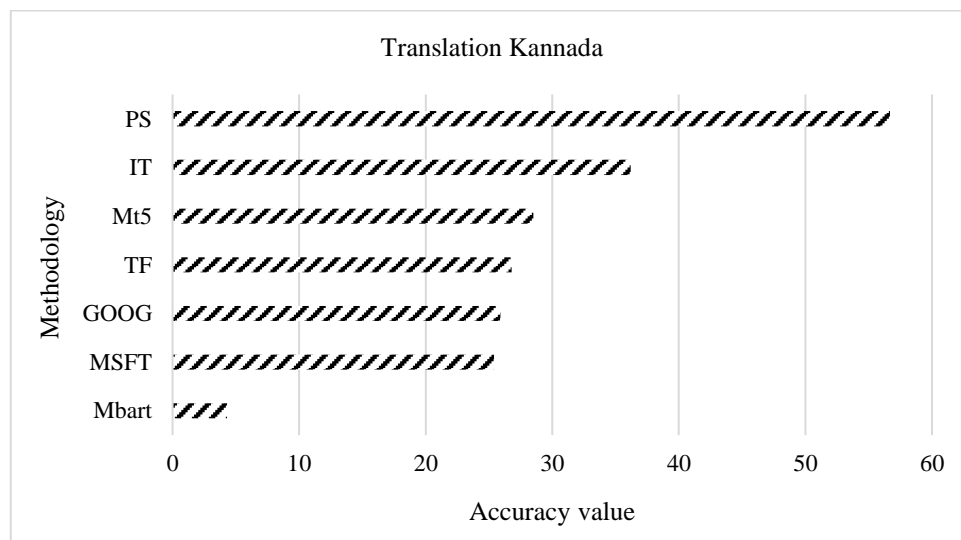


Figure 4. Accuracy comparison for translation in Kannada language

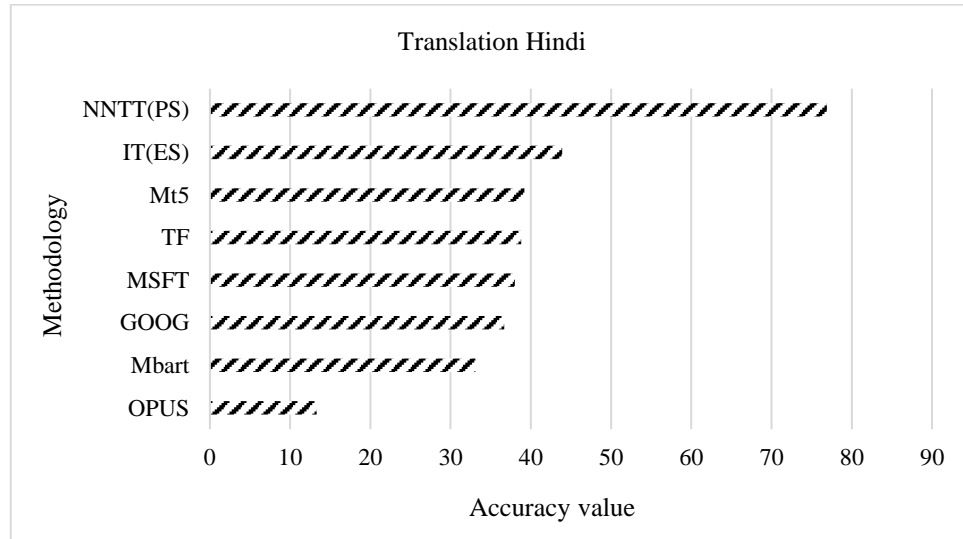


Figure 5. Accuracy comparison for translation in Hindi language

3.2. Comparative analysis

Table 1 shows a comparative analysis of the existing system and the proposed system for transliteration and translation tasks in Kannada and Hindi languages. The results indicate the percentage improvement achieved by the proposed system over the existing system. For transliteration in Kannada, the proposed system demonstrates a significant improvement of 10.66% in accuracy, achieving an impressive accuracy rate of 85.87%, compared to the existing system's accuracy of 77.18%. In the case of transliteration for Hindi, the proposed system outperforms the existing system by a remarkable 34.22%. The proposed system achieves an accuracy of 85.56%, while the existing system lags with an accuracy of 60.56%. Moving on to translation in Kannada, the proposed system shows remarkable progress, achieving an improvement of 44.12%. It attains an accuracy rate of 56.69%, while the existing system falls short with an accuracy of 36.2%. Similarly, for translation in Hindi, the proposed system excels with a remarkable 54.63% improvement. It achieves an impressive accuracy rate of 76.897%, while the existing system trails with an accuracy of 43.9%. The comparative analysis clearly demonstrates the superior performance of the proposed system in all evaluated tasks, showcasing its effectiveness in improving transliteration and translation accuracy for both Kannada and Hindi languages.

Table 1. Comparative analysis

Dataset	Existing system	Proposed system	Improvisation in (%)
Transliteration (Kannada)	77.18	85.87	10.6593
Transliteration (Hindi)	60.56	85.56	34.2185
Translation (Kannada)	36.2	56.69	44.1167
Translation (Hindi)	43.9	76.897	54.6322

4. CONCLUSION

In conclusion, this research introduces a novel SAEDM architecture that leverages the power of self-attention to construct an auto-encoder, avoiding the need for iterations or convolutional operations. The model exhibits exceptional performance in transliteration and translation tasks for Kannada and Hindi languages, achieving significant improvements over the existing systems. Incorporating lexical weighting and fine-tuning techniques, the proposed system efficiently trains at the word level, reducing training time while maintaining accuracy. Overall, the research displays the potential of the proposed SAEDM in advancing NLP tasks. The combination of self-attention, fine-tuning, domain adaptation, and binary classifier contributes to a robust and efficient system for handling diverse linguistic challenges. The proposed model, facilitates cross-cultural understanding, and improving the accessibility of information in diverse languages. As the world becomes increasingly interconnected, the effectiveness of such models in handling complex language tasks will play a crucial role in promoting effective cross-language communication and information access.

REFERENCES





- [1] Y. Zhao and H. Liu, "Document-level neural machine translation with recurrent context states," *IEEE Access*, vol. 11, pp. 27519–27526, 2023, doi: 10.1109/ACCESS.2023.3247508.
- [2] K. Mrinalini, P. Vijayalakshmi, and T. Nagarajan, "SBSim: a sentence-bert similarity-based evaluation metric for indian language neural machine translation systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1396–1406, 2022, doi: 10.1109/TASLP.2022.3161160.
- [3] A. Kumar, A. Pratap, and A. K. Singh, "Generative adversarial neural machine translation for phonetic languages via reinforcement learning," *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI)*, vol. 7, no. 1, pp. 190–199, Feb. 2023, doi: 10.1109/TETCI.2022.3209394.
- [4] S. Bhatia, A. Kumar, and M. M. Khan, "Role of genetic algorithm in optimization of Hindi word sense disambiguation," *IEEE Access*, vol. 10, pp. 75693–75707, 2022, doi: 10.1109/ACCESS.2022.3190406.
- [5] S. Saini and V. Sahula, "Neural machine translation for English to Hindi," in *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, IEEE, Mar. 2018, pp. 1–6. doi: 10.1109/INFRKM.2018.8464781.
- [6] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 154–163, Jan. 2020, doi: 10.1109/TPAMI.2018.2876404.
- [7] Y. Nishimura, K. Sudoh, G. Neubig, and S. Nakamura, "Multi-source neural machine translation with missing data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 569–580, 2020, doi: 10.1109/TASLP.2019.2959224.
- [8] H. Moon, C. Park, S. Eo, J. Seo, and H. Lim, "An empirical study on automatic post editing for neural machine translation," *IEEE Access*, vol. 9, pp. 123754–123763, 2021, doi: 10.1109/ACCESS.2021.3109903.
- [9] Y. Fan, F. Tian, Y. Xia, T. Qin, X.-Y. Li, and T.-Y. Liu, "Searching better architectures for neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1574–1585, 2020, doi: 10.1109/TASLP.2020.2995270.
- [10] O. Sen *et al.*, "Bangla natural language processing: a comprehensive analysis of classical, machine learning, and deep learning-based methods," *IEEE Access*, vol. 10, pp. 38999–39044, 2022, doi: 10.1109/ACCESS.2022.3165563.
- [11] J. A. Ovi, M. A. Islam, and M. R. Karim, "BaNeP: an end-to-end neural network based model for bangla parts-of-speech tagging," *IEEE Access*, vol. 10, pp. 102753–102769, 2022, doi: 10.1109/ACCESS.2022.3208269.
- [12] U. K. Acharjee, M. Arefin, K. M. Hossen, M. N. Uddin, M. A. Uddin, and L. Islam, "Sequence-to-sequence learning-based conversion of pseudo-code to source code using neural translation approach," *IEEE Access*, vol. 10, pp. 26730–26742, 2022, doi: 10.1109/ACCESS.2022.3155558.
- [13] Q. Du, N. Xu, Y. Li, T. Xiao, and J. Zhu, "Topology-sensitive neural architecture search for language modeling," *IEEE Access*, vol. 9, pp. 107416–107423, 2021, doi: 10.1109/ACCESS.2021.3101255.
- [14] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jan. 2016, doi: 10.18653/v1/N16-1101.
- [15] S. H. Sreedhara, V. Kumar, and S. Salma, "Efficient big data clustering using adhoc fuzzy c means and auto-encoder CNN," in *Lecture Notes in Networks and Systems*, Springer, Singapore, 2023, pp. 353–368. doi: 10.1007/978-981-19-7402-1_25.
- [16] F. Aqlan, X. Fan, A. Alqwbani, and A. Al-Mansoub, "Arabic-Chinese neural machine translation: romanized arabic as subword unit for arabic-sourced translation," *IEEE Access*, vol. 7, pp. 133122–133135, 2019, doi: 10.1109/ACCESS.2019.2941161.
- [17] S. A. Almaaytah and S. A. Alzobidi, "Challenges in rendering arabic text to english using machine translation: a systematic literature review," *IEEE Access*, vol. 11, pp. 94772–94779, 2023, doi: 10.1109/ACCESS.2023.3309642.
- [18] A. Sobhy, M. Helmy, M. Khalil, S. Elmasry, Y. Boules, and N. Negied, "An ai based automatic translator for ancient hieroglyphic language—from scanned images to English text," *IEEE Access*, vol. 11, pp. 38796–38804, 2023, doi: 10.1109/ACCESS.2023.3267981.
- [19] J. Zakraoui, M. Saleh, S. Al-Maadeed, and J. M. Alja'am, "Arabic machine translation: a survey with challenges and future directions," *IEEE Access*, vol. 9, pp. 161445–161468, 2021, doi: 10.1109/ACCESS.2021.3132488.
- [20] M. Shahroz, M. F. Mushtaq, A. Mehmood, S. Ullah, and G. S. Choi, "RUTUT: roman urdu to urdu translator based on character substitution rules and unicode mapping," *IEEE Access*, vol. 8, pp. 189823–189841, 2020, doi: 10.1109/ACCESS.2020.3031393.
- [21] S. Kausar, B. Tahir, and M. A. Mehmood, "ProSOUL: a framework to identify propaganda from online urdu content," *IEEE Access*, vol. 8, pp. 186039–186054, 2020, doi: 10.1109/ACCESS.2020.3028131.
- [22] K. Chen, R. Wang, M. Utiyama, and E. Sumita, "Integrating prior translation knowledge into neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 330–339, 2022, doi: 10.1109/TASLP.2021.3138714.
- [23] P. Pujar, A. Kumar, and V. Kumar, "Efficient plant leaf detection through machine learning approach based on corn leaf image classification," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, p. 1139, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp1139-1148.
- [24] Z. Tan, Z. Yang, M. Zhang, Q. Liu, M. Sun, and Y. Liu, "Dynamic multi-branch layers for on-device neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 958–967, 2022, doi: 10.1109/TASLP.2022.3153257.
- [25] G. H. Ngo, M. Nguyen, and N. F. Chen, "Phonology-augmented statistical framework for machine transliteration using limited linguistic resources," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 199–211, Jan. 2019, doi: 10.1109/TASLP.2018.2875269.
- [26] B. Roark *et al.*, "Processing south asian languages written in the latin script: the Dakshina dataset," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Jul. 2020.
- [27] G. Ramesh *et al.*, "Samanantar: the largest publicly available parallel corpora collection for 11 Indic languages," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 145–162, Feb. 2022, doi: 10.1162/tacl_a_00452.
- [28] J. Tiedemann and S. Thottingal, "OPUS-mt – building open translation services for the world," in *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, Lisboa, Portugal: European Association for Machine Translation, 2020, pp. 479–480.
- [29] Y. Tang *et al.*, "Multilingual translation with extensible multilingual pretraining and finetuning," *arXiv*, Aug. 2020, doi: 10.48550/arXiv.2008.00401.
- [30] M. Johnson *et al.*, "Google's multilingual neural machine translation system: enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, Nov. 2016.
- [31] A. Vaswani *et al.*, "Attention is all you need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long

Beach, CA, USA, Jun. 2017.





- [32] Y. Madhani *et al.*, “Aksharantar: towards building open transliteration tools for the next billion users,” *arXiv*, May 2022, doi: 10.48550/arXiv.2205.03018.
- [33] L. Xue *et al.*, “MT5: a massively multilingual pre-trained text-to-text transformer,” *arXiv*, Oct. 2020, doi: 10.48550/arXiv.2010.11934.

BIOGRAPHIES OF AUTHORS



Shanthala Nagaraja     earned her bachelor's of engineering (B.E.) degree in computer science and engineering from Kuvempu University in 2000. She has obtained her master's degree in (M.S.) software systems from BITS Pilani in 2006. Currently she is a research scholar at Department of Information Science and Engineering, Global Academy of Technology (Affiliated to VTU) doing her Ph.D. in computer science and engineering and also working as Senior Engineering Manager in Toshiba Software India Pvt Limited. Her areas of interest are natural language processing, deep learning, and internet of things. She can be contacted at email: shanthalatn@gmail.com or shanthala_12@rediffmail.com.



Dr. Kiran Y. Chandappa     earned his bachelor's of engineering (B.E.) degree from Kuvempu University. He has obtained his master's degree (M.Tech.) in software engineering from SJCE, Mysore in 2003 and Ph.D. from Jain University Bengaluru in 2015. He is currently working as Professor and Head of the Department of Information Science and Engineering, Global Academy of Technology, Bengaluru. He has worked at various verticals starting from lecturer to Professor and HOD. He worked as Member of Board of Studies and Board of Examinations of Visvesvaraya Technological University and many other neighboring universities and Autonomous Engineering Colleges. His research areas of interest include computer vision, pattern recognition, medical image analysis, data analytics, and blockchain. He can be contacted at email: kiranchandrappa@gmail.com.