INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

**Received:** 23-01-2023

**Accepted:** 01-04-2023

**Published:** 08-05-2023

**Citation:** Prasanna ASD, Latha CBC (2023) Bi-Lingual Machine Translation Approach using Long Short–Term Memory Model for Asian Languages. Indian Journal of Science and Technology 16(18): 1357-1364. https://doi.org/10.17485/IJST/v16i18.176

* **Corresponding author**.

adivyamca@gmail.com

**Funding:** None

**Competing Interests:** None

Published By Indian Society for Education and Environment (iSee)

**ISSN**
Print: 0974-6846
Electronic: 0974-5645

# Bi-Lingual Machine Translation Approach using Long Short–Term Memory Model for Asian Languages

**A Shalini Divya Prasanna[1]*, C Beulah Christalin Latha[2]**

**1** Research Scholar, Department of Digital Sciences, Karunya Institute of Technology and Sciences, Coimbatore, 641114, Tamil Nadu, India
**2** Associate Professor, Department of Digital Sciences, Karunya Institute of Technology and Sciences,, Coimbatore, 641114, Tamil Nadu, India

## Abstract

**Objectives**: To develop an appropriate machine translation model for translating text from English to Tamil. **Methods:** The proposed work uses a Gated Recurrent Unit (GRU) Long Short-Term Memory (LSTM) model. The Repeat Vector function is used for fitting both the decoder and encoder parts of the network model. Adam optimizer is used because of its faster execution and less consumption of memory. It mainly uses the text corpora which are available in the Internet repository namely Technology Development for Indian Languages (TDIL), Linguistic Data Consortium for Indian Languages (LDCIL), Kaggle, and Ishikahooda. **Findings:** The motivation for the proposed work emerged from identifying the regional language Tamil as one of the less frequently used languages in the existing translation systems. The Tamil Character Set is one of the challenging factors for the existence of fewer such translation systems. The proposed system produces a BLEU score of 0.9, a Meteor score of 0.98, a TER score of 0.5, a WER score of 20%, an Accuracy rate of 5 (in a 5-point grading scale), and an Adequacy rate of 5 (on a 5-point grading scale) which are significantly better than the existing systems. **Novelty:** The space complexity of the proposed LSTM-based English Tamil Translator is fine-tuned to 256 units of memory using Adam optimizer for achieving less storage consumption. The number of layers is optimized for reducing the execution time. Unicode Transformation Format (UTF-8) encoding is used to incorporate Tamil language characters. This work has been implemented with a wide range of sentences counted to several thousand. LSTM-based English Tamil Translator is helpful for bilingual learners who are learning specifically Tamil language.

**Keywords:** Machine translation; Deep Learning; LSTM; English; Tamil

# 1 Introduction

Machine translation is an interdisciplinary research domain encompassing various fields such as Artificial Intelligence, Natural Language Processing, Mathematics, and Statistics. Such kind of translation services provided by machine translation systems is highly appreciable in this scenario[(1–3)]. The translation tasks were primarily handled using classical and statistical methodologies. However, the classical approach was not preferred by many due to its pitfalls. The main disadvantages are the rules developed by the expert and the huge number of requirements of strategies and exemptions. The data-driven approach, namely the statistical approach, outperforms the performance of the classical approach[(4–6)]. This technique also eliminates the necessity of a linguistic expert and requires only a text corpus of both the source and target language examples. But, the results are superficial for languages with different word orders. Deep Neural Machine Translation models are being considered for a wider range of translation tasks in recent times. LSTM uses memory cells for retaining the values and also requires a minimal amount of training data. Hence, the proposed system overcomes the existing systems by using the NMT model involving GRU LSTM in both the encoding and decoding phases[(7–9)]. The proposed work has been implemented using a large number of sentences counted to several thousand. Both the human subjective scores (Adequacy, Fluency, Accuracy) and automated scores (BLEU, Meteor) have been used for analyzing the results[(10–12)].

## 1.1 Literature Review

An exhaustive review has been conducted on the existing machine translation models that are used in various language conversions and comparisons[(5,13–17)] . A neural machine translation system is developed for the conversion of sentences from Marathi to English. The Byte Pair Encoding algorithm is used for representing the input sequences. LSTMs are used in the existing systems for achieving the required translation[(5)] . A machine learning-based translation model is developed for converting the text from Hindi to English. The pattern recognizer using quantum neural is incorporated into recognizing and learning the corpus patterns. The existing system thus performs the machine translation using sentence pairs of Devanagari-Hindi and English. The dataset of 2600 sentences has been used for the implementation of the system[(13)]. A model for converting English text to two Indian languages such as Tamil and Punjabi is developed by using the NMT. This model includes both the attention mechanism and the score function for an effective translation. Human evaluators evaluated the quality of the translated output using the Bi-Lingual Evaluation Under Study concerning the fluency and adequacy of the predicted output[(14)]. A transformer-based NMT model is developed for the translation of English sentences to eight Asian languages. The model highlights the layer normalization component for the convergence of different translations. The combination of such eight translation approaches is carried out in the multilingual NMT model[(15)].

A framework has been suggested for the neural machine algorithm for the translation of English resources. The model includes the combination of both neural machine and statistical machine translation for the effective translation of words. The vocabulary alignment structure is used by the decoder to reduce vocabulary-related problems[(16)]. A classical translation system known as the English-to-Bengali MT system is developed for the translation between English and Bengali languages. The system performs well in some circumstances, but a larger corpus makes it more difficult to build a large database. This system was created with a tiny corpus. Due to the structural similarities between Urdu and Hindi, an English-to-Urdu machine translation system is developed by using Hindi as a medium of conversion. Hindi is first translated from the input English sentences, and then Hindi is translated into Urdu. This system uses Interlingua and rule-based methodologies. The Urdu term was mapped to the matching Hindi word using the Hindi-Urdu mapping database. According to industry standards, the system's BLEU score for translating from English to Urdu is 0.3544[(17)].

## 1.2 Research Gap

It has been observed that some of the existing systems were developed through classical and statistical methods[(5,13–17)] . The existing research works on language translation are found to be very minimal for the conversion of the text from English to Tamil. This developed the intuition for the proposed system and an exhaustive analysis has been made of a few such existing systems. The Space complexity of the proposed LSTM-based English - Tamil Translator is fine-tuned to 256 units of memory. The number of layers is optimized for achieving less Time complexity in generating the desired output. Unicode Transformation Format (UTF-8) encoding is used to assist their learning and translation processes. Even a few systems that were developed using NMTs consist of a low number of sentences. This work supports bilingual learners especially in learning the Tamil language with a wide range of sentences counted to several thousand.

## 2 Methodology

The proposed LSTM-based English - Tamil Translator is implemented using the LSTM model. Figure 1 depicts the outline of the proposed methodology.

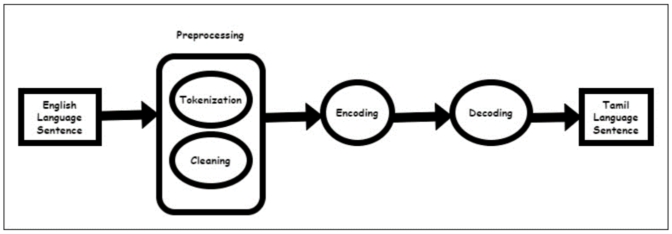

**Fig 1.** Structure of LSTM-based English - Tamil Translator

A detailed explanation of the working of the LSTM-based English - Tamil Translator system is explained is provided in the following section:

### 2.1 Preprocessing

Pre-processing of the text corpus is considered to be the most important phase in developing an NMT system. It is a 2-step process that includes Tokenization and Cleaning. Tokenization is performed on the sentences to divide them into words separated by white spaces. It is done by using Keras API for the text corpus of the proposed system. Cleaning is helpful to remove the non–alphabet, special characters, unnecessary spaces, and improper sentences from the text corpus. The input sentences are then cleaned up so that the existence of any special characters can be removed for further processing. The text corpus is loaded in the UTF -8 format for the proposed system because UTF-8 can efficiently store text containing any human language character.

### 2.2 Encoding

Encoding is done by the Encoder of the LSTM-based English - Tamil Translator system. It acquires only one grapheme or word in a timestep. Hence, for the input sentence of m words or length L, L time steps are taken to read it. Each word is then mapped into the index of the language vocabulary for creating a thought vector or context vector. Context Vector represents the meaning of the sentence in the source language. converted into a fixed-sized vector. The task of this component is to perform the proper UTF-8 and ASCII encoding i.e., American Standard Code for Information Interchange encoding of the input sentences. Considering S as a sentence in the native source language (English) and T as its corresponding sentence in the target language (Tamil). The encoder converts S ($s_1$, $s_2$, $s_3$..., $s_m$) into fixed dimension vectors which in turn are foreseen literally by the decoder using the conditional probability that is given in Eq. (1).

$$P(T/S) = P(T \mid S_1,\ S_2,\ S_3, \ldots, S_M) \tag{1}$$

The following are the notations that are used such as $x_t$ is the input at time step $t$; $h_t$ and $c_t$ are the hidden states of LSTM at the time step $t$; $y_t$ is the output produced at time step $t$. Considering a sentence in the source language English such as "I slept", the same is interpreted as a sentence consisting of two words namely, $s_1$ = "I", and $s_2$ = "slept" in the proposed system. So, this sequence is read in two - time steps as shown in Figure 2 .

At t = 1, it remembers "I," and when time t = 2, it recalls that the LSTM has read "I slept," and the states h2 and c2 remember the complete sequence "I slept".

The initial states such as $h_0$ and $c_0$ are initialized as zero vectors. The encoder takes the above sequence of words X ={$x_s^1$, $x_s^{2,}$ $x_s^3$,.... $x_s^L$ } as the input and calculates the thought vector v = {$h_c$, $v_c$}, where $h_c$ represents the final external hidden state which is obtained after processing the final element input sequence and $v_c$ is the final cell state. It can be mathematically represented as $v_c = c_{L,}$ and $v_h = h_L$.

Here, $S_1$, $S_2$, ..., and $S_M$ represent the fixed-size encoded vectors. Eq. (1) is then converted to a form as written in Eq. (2), by using the chain rule.

$$P(T \mid S) = P(t_i \mid t_0, t_1, t_2, \ldots, t_{i-1}; s_1,\ s_2,\ s_3, \ldots, s_m) \tag{2}$$

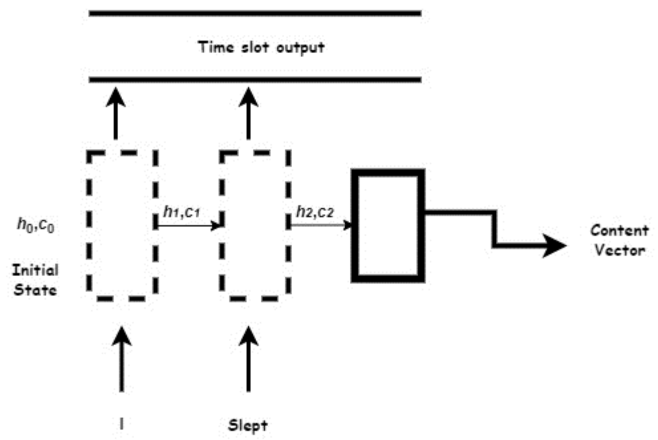

**Fig 2.** Sentence reading by Encoder

These kinds of NMT models that are developed using the above Eq. (2) are referred to as Left – to – Right (L2R) autoregressive NMT. The decoder predicts the next word by using the previously determined word vectors and also the source sentence vectors which are given in Eq. (1).

## 2.3 Decoding

The decoding is done by the decoder of the LSTM-based English - Tamil Translator system. The role of this is to decode the context vector into the desired translation. The parameters that are used to initialize the decoder are the context vector v = {$v_h$, $v_c$} represented as $h_0 = v_h$ and $c_0 = v_c$. At each step of decoding, a decoder uses the global attention mechanism by checking against the entire pool of source states which is shown in Figure 3.

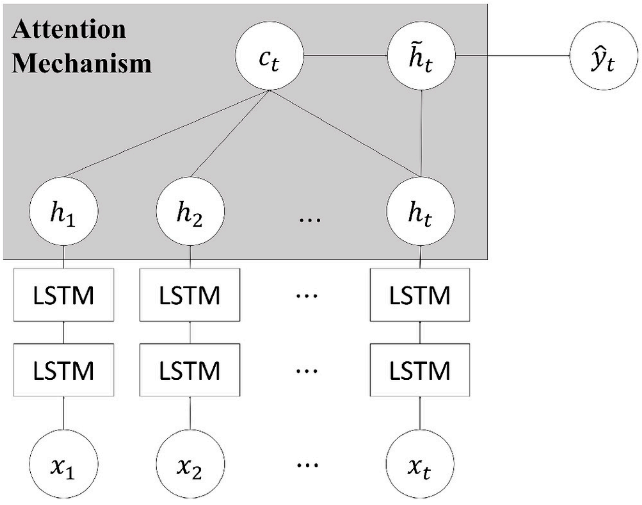

**Fig 3.** Attention Mechanism

The m<sup>th</sup> prediction of the translated sentence is determined by the Eq. (3).

$$\begin{gathered} C_m, h_m = \left\{y^1 T, y^2 T, \ldots, y^{M_T}\right\} \\ y_m^T = \operatorname{softmax}\left(w_{\text{softmax}} * h_m + b_{\text{softmax}}\right) \end{gathered} \tag{3}$$

## 3 Results and Discussion

The proposed LSTM-based English - Tamil Translator system mainly uses the text corpora which are available in the Internet repository namely TDIL, LDCIL, and manythings.org. It contains around three thousand Tamil words and their English equivalent words from a repository of more than two hundred sentence pairs. This system is implemented in the TensorFlow platform. The following Table 1 shows the description of datasets used in the proposed system.

**Table 1.** Datasets used in the LSTM-based English - Tamil Translator

| Datasets | Total Number of English Sentences | Total Number of Tamil Sentences |
|---|---|---|
| Technology Development for Indian Languages (TDIL) | 6500 | 6500 |
| Linguistic Data Consortium for Indian Languages | 5000 | 5000 |
| Manythings.org | 2000 | 2000 |
| **Kaggle** | **7000** | **7000** |
| Ishikahooda | **236,427** | **236,427** |

The stepwise implementation of the language-translation is described as follows: An encoder-decoder LSTM model is used for the proposed English-to-Tamil translation system. The size of the English and Tamil languages' vocabularies, as well as the overall amount of memory units, used to hold both the encoded and decoded words, are all significant characteristics that are used to configure the model. The complexity of the translation is usually predicted using the parameters such as the count of sentence pairs available in the data set, the length of each such pair, and the size of the vocabulary. The training and testing data set is used in a ratio of 70:30 respectively. The language tokenizers also figure out how much vocabulary is used in each language and how long a sentence may be for a given language phrase. Figure 4 represents the proposed NMT model.

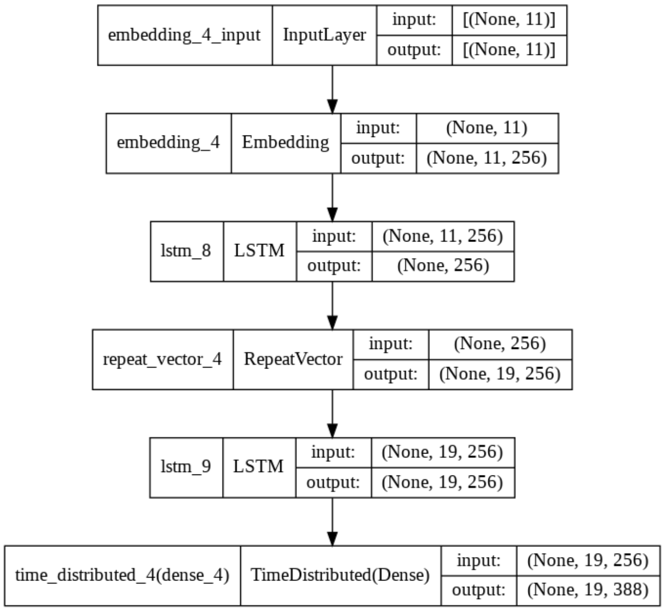

**Fig 4.** Proposed LSTM-based English - Tamil Translator Model

The language model used in the proposed system is represented in Table 2 .

**Table 2.** Language Model of the LSTM-based English - Tamil Translator

| Size of the vocabulary of the languages | Total number of words used |
|---|---|
| Source language | 388 |
| Target language | 599 |

LSTM-based English - Tamil Translator uses an encoder-decoder-based LSTM model that encompasses the sequence of inputs which is encoded by a front-end model known as the encoder.

i) Preprocessing source sentence $x_s = x_1, x_2, x_3, \ldots, x_L$, and target sentence $y_t = y_1, y_2,, y_3, \ldots, y_L$ pairs;

ii) Performing embedding using the embedding layer;

iii) Passing $x_s$ into encoder;

iv) Computing content vector v across the attention layer conditioned on $x_s$.

v) Setting $(h_0c_0)$ of the content vector by using the initial states of the decoder;

vi) Predicting target sentence $y_T = \{y^1{}_T, y^2{}_T, ..., y^M{}_T\}$ for the input sentence $x_{s,}$ where m$^{th}$ prediction in the target vocabulary is determined by using

$y^M{}_T$ = softmax $(w_{softmax}h_m + b_{softmax})$;

vii) Determining the loss using categorical cross entropy for the words between the predictions and the actuals at the m$^{th}$ position;

viii) Optimizing encoder and decoder by updating the weight matrices (W, U, V) and softmax layer;

Input is utilized by the encoder to produce a fixed-sized vector from it. By using both the attention mechanism and the scoring function, the decoder assesses the possibility of discovering a potential target word related to the source word. The target embedding involves the repeat vector functionality to generate the target Tamil language words concerning the input. The Repeat Vector layer involves the maximum word length of the Source language as specified in Table 1 , along with 256 units. The Target Embedding layer then uses the above parameters along with the maximum word length of the Source language as specified in Table 1 . The total number of layers that are hidden and output layers are represented in Table 3 .

**Table 3.** Representation of the Layers of the LSTM-based English - Tamil Translator

| Layer (type) | Output Shape | Param # |
|---|---|---|
| embedding | (None, 11, 256) | 143104 |
| lstm | (None, 256) | 525312 |
| repeat_vector | (None, 19, 256) | 0 |
| lstm | (None, 19, 256) | 525312 |
| time_distributed | (None,19,388) | 99716 |

The Input layer is defined by using the maximum length of the target language i.e., the Tamil word that is used in the proposed system. The maximum length of the phrases of the generated output in the target language Tamil and the total amount of memory units required to configure the suggested model is utilized to generate the Embedding and LSTM layers. The Embedding and LSTM layers are created using the maximum length of the phrases of the generated output in the target language, and the total amount of memory units needed to design the suggested model. The Repeat Vector layer is similar to an adapter. It is used for fitting both the decoder and encoder parts of the available network model. This layer simply repeats the given 2D input several times to create a 3D output. The Time Distributed wrapper reuses the same output layer for each element in the output sequence. The maximum word length allowed in the source language, English, as well as the total amount of memory units needed to configure the suggested model, is used to generate this wrapper.

## 3.1 Evaluating the proposed model

The model is evaluated on both the train and test datasets. This involves two steps such as i) generating the output iteratively for many input sets of data and ii) summarizing the model generation according to step i). Both the forward and backward mapping of converting words to numbers and numbers to words are carried out simultaneously using this model in translating the source to the target language(18).

### *3.1.1 Automatic and Human Evaluation Scores*

Human subjective evaluations of some aspects of the output, such as fluency in a 5 -point grade scale (with the highest score of 5) and adequacy, are used as human intrinsic measurements to establish quality. Automatic intrinsic measurements compare the corresponding MT output against a predetermined set of reference translations to create rankings among MT systems using an easily calculated phrase similarity measure such as Meteor, TER (Translation Edit Rate), and WER (Word Error Rate)(14). The above results are summarized and shown in Table 4.

**Table 4.** Summary of the results of the proposed LSTM-based English - Tamil Translator

| Dataset | Adequacy | Fluency | Accuracy | BLEU | Meteor | TER | WER |
|---|---|---|---|---|---|---|---|
| TDIL | 3 | 4 | 5 | 0.99 | 0.98 | 0.47 | 17% |
| LDCIL | 3 | 5 | 4 | 0.94 | 0.96 | 0.50 | 20% |
| Manythings.org | 3 | 5 | 5 | 0.96 | 0.97 | 0.50 | 19% |
| **Kaggle** | **4** | **4** | **5** | **0.98** | **0.97** | **0.48** | **18%** |
| Ishikahooda | **3** | **5** | **5** | **0.97** | **0.96** | **0.49** | **19%** |

### *3.1.2 Comparison with the existing systems*

The Bi-Lingual Evaluation Under Study or, otherwise BLEU score is used for the comparison of the existing systems. The BLEU score is used in the evaluation of the machine translations for the different languages. It is determined using the number of words generated in the machine translation output that are matching with the reference translation[5,13–17] . The proposed system is found to achieve a score of 9 which is found to be better than the available translation systems. A comparison is made against the observed BLEU scores of the proposed and existing systems are shown in Figure 5 .

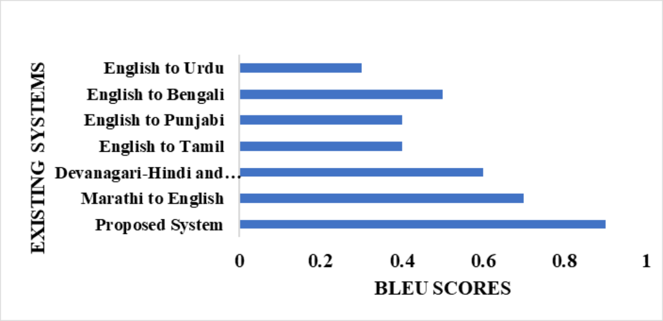

**Fig 5.** Comparison of BLEU Scores

From the above Figure 5 , it is inferred that the LSTM-based English - Tamil Translator can perform more efficiently than the existing translation systems which were adopting the NMT for converting between the English and Tamil languages.

## 4 Conclusion

This study concludes that the GRU LSTM model is the best method implemented for the proposed English-to-Tamil bilingual translation system. The LSTM-based English - Tamil Translator model can perform the required translation effectively by achieving less time and space complexity. This translator is very supportive of bilingual learners as it is implemented using huge datasets. The results are evaluated by using both the human subjective and the machine evaluation scores and are found to be better than the existing systems. Future research should widen its scope to support other Indian languages using Tamil as their primary language.

## Acknowledgment

The Tamil language text datasets used in the proposed research work are being maintained by a) Technology Development for Indian Languages (TDIL), a Programme initiated by the Ministry of Electronics & Information Technology, Government of India and b) Linguistic Data Consortium for Indian Languages (LDCIL), scheme of the Department of Higher Education, Ministry of Human Resource and Development, Government of India. The datasets were shared and accessed free of charge on request.

## References

1) Syed A, Abdul W. Machine Translation System Using Deep Learning for English to Urdu. 2022. Available from: https://doi.org/10.1155/2022/7873012.
2) Laith A, Jinglan Z, Humaidi AJ, Ayad AD, Ye D, Omran AS, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*. 2021;8:53–53. Available from: https://doi.org/10.1186/s40537-021-00444-8.

3) Antonio HB, Boris HF, David T, Borja NC. A Systematic Review of Deep Learning Approaches to Educational Data Mining. *Hindawi Complexity*. 2019. Available from: https://doi.org/10.1155/2019/1306039.
4) Kumar MA, Premjith B, Singh S, Rajendran S, Soman KP. An Overview of the Shared Task on Machine Translation in Indian Languages (MTIL) – 2017. *Journal of Intelligent Systems*. 2019;28(3):455–464. Available from: https://www.degruyter.com/document/doi/10.1515/jisys-2018-0024/html.
5) Mandke A, Litake O, Kadam D. Analyzing Architectures for Neural Machine Translation using Low Computational Resources. *International Journal on Natural Language Computing*. 2021;10(5):9–16. Available from: https://doi.org/10.5121/ijnlc.2021.10502.
6) Karyukin V, Rakhimova D, Karibayeva A, Turganbayeva A, Turarbek A. The neural machine translation models for the low-resource Kazakh–English language pair. *PeerJ Computer Science*. 2023;9:e1224. Available from: https://doi.org/10.7717/peerj-cs.1224.
7) Akshai R, Venkatesh BP, Rejwanul H, W A. Comparing Statistical and Neural Machine Translation Performance on Hindi-To-Tamil and English-To-Tamil. 2021. Available from: https://doi.org/10.3390/digital1020007.
8) Premjith B, Soman KP. Deep Learning Approach for the Morphological Synthesis in Malayalam and Tamil at the Character Level. *ACM Transactions on Asian and Low-Resource Language Information Processing*. 2021;20(6):1–17. Available from: https://doi.org/10.1145/3457976.
9) Zhixing T, Shuo W, Zonghan Y, Gang C, Xuancheng H, Maosong S, et al. Neural machine translation: A review of methods, resources, and tools. 2020. Available from: https://doi.org/10.48550/arXiv.2012.155.
10) Vijaay KU, Mahesh N, Karthikeyan B, Rama S. Attention based Neural Machine Translation for English-Tamil Corpus. *International Research Journal of Engineering and Technology (IRJET)*. 2020;p. 4–4. Available from: https://www.irjet.net/archives/V7/i4/IRJET-V7I4703.pdf.
11) Kandimalla A, Lohar P, Maji SK, Way ASK. Improving English-to-Indian Language Neural Machine Translation Systems. *Information*;13(5):245. Available from: https://doi.org/10.3390/info13050245.
12) Harish BS, Rangan RK. A comprehensive survey on Indian regional language processing. *SN Applied Sciences*. 2020;2(7). Available from: https://doi.org/10.1007/s42452-020-2983-x.
13) Narayan R, Chakraverty S, Singh VP. Quantum neural network based machine translator for English to Hindi. *Applied Soft Computing*. 2016;38:1060–1075. Available from: https://doi.org/10.1016/j.asoc.2015.08.031.
14) Amarnath P, Partha P. Neural Machine Translation for Indian Languages. *Journal of Intelligent Systems De Gruyter*. 2019;28(3):465–477. Available from: https://doi.org/10.1515/jisys-2018-0065.
15) Raphael R, Benjamin M, Raj D, Atushi F, Masao U, Eiichiro S. Extremely low-resource neural machine translation for Asian languages. 2020. Available from: https://doi.org/10.1007/s10590-020-09258-6.
16) Yanping Y. Translation Mechanism of Neural Machine Algorithm for Online English Resources. *Hindawi Complexity*. 2021. Available from: https://doi.org/10.1155/2021/5564705.
17) Andrabi S, Wahid A. A review of machine translation for South Asian low resource languages. *Turkish Journal of Computer and Mathematics Education*. 2021;12(5):1134–1147. Available from: https://pdfs.semanticscholar.org/d9d9/9880f50dc1cf7ff58a40869efdfa9e723d09.pdf.
18) Ramesh A, Parthasarathy VB, Haque R, Way A. Comparing Statistical and Neural Machine Translation Performance on Hindi-To-Tamil and English-To-Tamil. *Digital*. 2021;1(2):86–102. Available from: https://doi.org/10.3390/digital1020007.