English To Hindi Translator Using Seq2seq Model

Sindhu[1] C[0000-0001-9899-0046],Soumyajit Guha[2],Yuvraj Singh Panwar[3]
[1,2,3]*Department of Computing Technologies, School of Computing*
SRM Institute of Science and Technology, Kattankulathur,India - 603203
sindhucmaa@gmail.com

2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) | 978-1-6654-0816-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICACCS54159.2022.9785158

***Abstract*—Research in Machine Translation has been going on since the 1950's. It began in the United States.Since then, we have come a long way in this field of research.Various algorithms have been derived and tested to fulfill the requirements for an ideal translation machine which functions in all the languages which are spread throughout the world. Any machine translation system's success depends on a solid translation algorithm. If there is a large quantity of space available, decoding will be sluggish, which is undesirable. There is a trade-off between the precision of the translation and the speed with which it is decoded. Basically, what the machine learning-based approach proposed does is to identify the input words and categorize them under a few metrics after which they are translated and rearranged to form the output. We then present the output in the desired language.**

Keywords--Natural Language Processing, Recurrent neural Network, LSTM, Sequence-to-Sequence Model, Encoder-Decoder Model

I. INTRODUCTION

The need for translation is extremely vital in today's world. With so many languages all over the world and with the advancing technology it is extremely important for us to minimize the lingual barrier among us all. For example, take the scenario that a Japanese scientist discovers some theorem or thesis and his research is in his native language. Now if someone from the USA has some interest in that work, then he first needs to translate his work in English and only then can he work on it further.

Now the problem with the existing solutions to this problem statement is that the translation isn't perfect when translating sentences. If only word translations are considered then it works pretty fine, but when translating sentences, the meaning of the sentence gets lost within the translation.

Natural Language Processing (NLP) [3] is a way in computer science to derive and analyze human language and interact with humans in a useful and smart way. Using NLP, we intend to create a machine translation system which with minimal errors translates English to Hindi.

The words and phrases of human languages are not immediately understood by computers. Binary numbers are only understood by the computer as 0s and 1s. As a result, we had to first devise a method for computers to comprehend the language. For this problem, word representation is a frequently used approach. The technique of representing a word with a vector is known as word representation, and each word has its own vector representation.

Text and word representation are necessary for computers to understand words thus, we must encode words in a way that computers can understand. One approach for converting categorical data to numerical data is one-hot encoding. The algorithms then learn and forecast using the numerical data.

As previously said, machine translation translates meaningful content from one language to another without the use of humans. The BLEU (Bilingual Evaluation Understudy) score is used to assess machine translation. BLEU is a statistic for evaluating machine-translated text automatically. The score ranges from 0 to 1, with a higher value indicating better machine translation.

II. RELATED WORK

For decades, machine translation has been a part of the industry. There have been initiatives to accomplish automated translation since the 1970s. Three key methods have evolved throughout time:

TABLE I. RELATED WORK AND ITS TASKS

Paper Id	Task Performed	Unexplored Fields
[7]	Hybrid MT is builtusing a mixture of RBMT and SMT, new rules are delivered to the system for betterdegree of efficiency.	model cannot be used for complex sentences
[13]	Chinese-Uyghur machine translation is a tough projectdue to the complexity of the morphological and syntactic systems of Uyghur, consequently NMT is used.	Smaller dataset and corpus are used which offers low accuracy and POS tagging and phrase correspondence among Uyghur and Chinese have to have been accomplished with assist of dictionaries
[18]	Word-based SMT knowledge is provided to Neural Machine Translator to improve the performance	word-based SMT were used which has low accuracy, in preference to s phrase-based totally SMT

A. *Rule-based Machine Translation (RBMT):*

RBMT [5] systems are based on linguistic concepts that allow words to be used in different contexts and have different meanings. The RBMT method is used to apply language rules such as transfer, analysis, and creation[6]. These rules are created by human language specialists and programmers.Advantages are that you do not need bilingual text. Complete control (new law applicable in all cases). Reusable (existing language rules can be transferred when paired with new languages). And the disadvantages are that they need good dictionaries. Manually set rules (requires expertise)

B. *Statistical Machine Translation (SMT):*

Statistical Machine Translation (SMT) [3] is mostly trained on bilingual text corpus, which is a collection of existing human translations. As we've seen, the RBMT system is primarily concerned with word-based translation, whereas the SMT system is concerned with phrase-based translation. Phrase-based translation aims to overcome the limitations of word-based translation by translating complete sequences of words of various lengths. The word sequences are referred to as phrases, however they are usually not linguistic phrases, but rather phrases identified by statistical methods in bilingual text corpora.

C. *Present Neural Machine Translation (NMT)*

NMT is a well-known and extensively used translation service that uses an end-to-end approach to automatic translation to solve the shortcomings of RBMT and SMT. NMT produces superior translation output than other standard Machine Translation systems by utilizing the most modern deep learning algorithms. It is the most modern kind of machine translation that uses a neural network that is closely akin to the neurons in the human brain to categorize input into different groups and layers.NMT [4] is a method of language translation that focuses on the context of sentences or paragraphs rather than single words. The NMT system consists of up-to-date multilingual databases as well as automatic learning processes that aid in continual improvement.

Benefits of Machine Translation [16] are Time efficient: Machine Translation models can save a significant amount of time as they can translate a whole document in sec. Cost efficient: It does not require human involvement which leads to lower cost. Memorizes terms: Machine Translation models are designed in such a way that they memorize the key terms and reuses them wherever they fit. When it comes to natural language processing, sky's the limit. The future will witness some huge alterations in this subject as technology becomes more prevalent and further improvements are explored.

III. METHODOLOGY

We built a Machine translation model using a neural network and Sequence-to-Sequence model. The overall technique has been accomplished in four basic steps: segmentation, tagging, translation, and rearranging.

A. *Dataset:*

We utilized a Kaggle dataset "English to Hindi Neural Machine Translation". Text preprocessing is the first step in the process of building a model. Datacleaning is a completelyvital step in any systemstudyingversion, howevergreater so for NLP. without the cleaningsystem, the dataset is usually a cluster of words that the computer doesn't recognize.

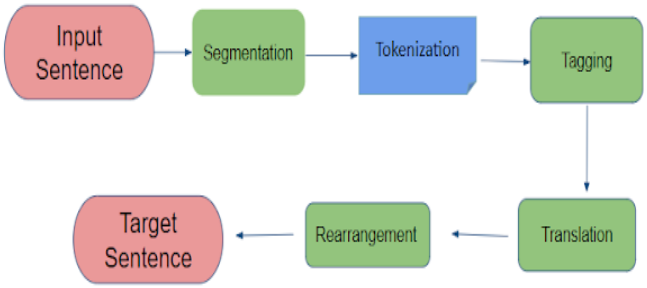

Fig. 1 - English to Hindi Translator flow diagram

B. *Segmentation:*

Tokenization is the first step in segmentation, in this step the main task is to divide the text into groups or "tokens" as the name suggests. This is an important step as the machine needs to act differently on different tokens. As a result, proper tokenization of the text is critical. Tokenization is the process of breaking down a statement into individual words. All the machine does in this step is to make sure to bifurcate the sentence in such a way that the individual parts of the sentence can be targeted by the translation algorithm in a similar manner and the process is carried on smoothly and efficiently.

Example:
The house number is 321. → here we have to split "." with "321". M.B.B.S. → Special Case: Abbreviation

C. *Tagging:*

In this step the numbers and names, i.e., the proper nouns in the input sentence are marked with placeholders. This is helpful as now when the machine will come across such placeholders it will not perform any translation process on it. Similarly, numbers will also be tagged by placeholders so as to contain any translation which includes meaning translation isn't carried out on it and only required steps are carried out on it. After this step is carried out the machine gets a lot of clarity on how to apply the machine translation on the target sentence so as to get a reasonable result.

Example:
My name is May. Here the word "May" gets tagged as a proper noun in this sentence. The house number is 321. → tagging number 321 Therefore it becomes → The house number is <number>.

D. Translation:

Understanding the meaning of the English words and translating them accordingly to Hindi. The translation should be done with maximum accuracy. This accuracy is obtained after the machine is trained on the basis of a diverse dataset which is fed to the machine beforehand. The more diverse dataset is used to train the machine, the more accurate the machine will translate the input sentence. This is because by having more dataset to train on the machine gets a wide variety of sample data to understand the grammatical nuances of the language and how the words are arranged so as to give a meaningful output.

Example:

How are you. → "how", "are", "you" → "आप", "कैसे", " हैं". My name is May. → "My", "name", "is", "May" → "मेरा ", "नाम ", "हैं", "में"

Recurrent Neural Network: Recurrent Neural Network (RNN)s are a branch of artificial intelligence that is stable, durable, and they are one of the most promising algorithms in use due to the reason that they are the only ones with only an internal memory. Similar to other deep learning approaches,RNN [9] is quite new. They were initially developed in the 1980s, but we didn't realize their full potential until recently. Increased computer power, bigquantities of data we now have to deal with, and the introduction of long short-term memory in the 1990s have all contributed to the emergence of RNNs. The recurrent neural network [8] is mostly utilized for problems involving natural language processing.Word prediction/auto completion, machine translation, name entity recognition, and sentiment analysis are all applications of RNN. Simply said, recurrent neural networks are better at anticipating sequential input than other algorithms. As an increasing number of records piles up RNN turns into much less powerful at learning new things.

Benefits of a RNN are that An RNN remembers every piece of information throughout time. It can only be used to predict time series since it remembers previous inputs. Long Short-Term Memory is the term for this type of memory (LSTM).[10] With recurrent neural networks, even convolutional layers are used to extend the effective pixel neighborhood. Disadvantages of RNN are that Traditional RNNs have short-term memory because of the vanishing gradient problem. It's difficult to train an RNN. Tanh or relu will not be able to handle excessively lengthy sequences when used as an activation function.

Long-Short-Term-Memory: To be more exact, we employed Long-Short-Term-Memory(LSTM), a better version of recurrent neural network, in our machine learning technique. In sequence prediction problems, LSTM [13] networks are a type of recurrent neural network that can learn order dependency.LSTM solves the short time period memory loss problem that arises in RNN and facilitates to get higher predictions because it lets in a neural network to take into account the stuff it needs and overlook the stuff not required.This is a necessity in a number of difficult problem areas, such as machine translation and speech recognition, among others. LSTM [19]has been created to reduce the vanishing and exploding gradient problem. LSTM is divided into 4 components: (1) Input gate, (2) Memory cell, (3) Forget gate, (4) Output gate. A memory cell is a type of memory cell that is utilized to remember and forget things. The context of the input influences how we remember and forget things. When we give input from here in a pointwise process, the Forget Gate forgets some of the information while keeping some of it in the function. Information is stored in the memory cell. Based on input and then passed to the output gate. Further we have used sequence to sequence encoder decoder models for our machine.

Encoder-decoder Model: A sequence-to-sequence model, first introduced by Google in 2014,where a model takes a sequence of information and outputs a sequence of information in which the size of the input sequence and the output sequence can also differ.i.e. The encoder/decoder architecture is a type of architecture that takes in an input sequence and it may produce a sequence of varied lengths from the input.

For example,when translating "What is your name?" from English to Korean, for example, the input is 4 words and the output is 6 symbols (이름이 뭐예요). Clearly, we won't be able to cover each word from the English sentence to the Korean language using a standard LSTM network.

This is why, in situations like those, the sequence-to-sequence approach is employed. In order to understand the model's logic, we will look over the illustration below: The encoder, intermediate (encoder) vector, and decoder are the three components of the model.

Encoder: A stack of LSTM cells that each receives an element of the enter sequence, collects info for that element, and propagates it ahead. Every input textual content is passed within the form of a vector to the encoder at a timestamp *t* from a sequence of inputs, and the encoder provides an output inside the form of a context vector.

In a question-answering issue, the enter collection is a set of all phrases from the inquiry. The sign x_t is used to symbolize each word, with *t* indicating the word's order. The formula (1) is used to calculate the hidden states h_t:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

This simple formula shows the outcome of a regular RNN. We simply practice the required weights to the previous hidden state h_{t-1} and the input vector x_t, as you can see in (1).

Encoder Vector: This is the model's final hidden state, created by the encoder. It is calculated using the formula above. This vector aims to include all input element information to assist the decoder to make correct

predictions. For the decoder, it is the model's initial hidden state.

Decoder: A set of recurrent units, each of which anticipates an output y t at a given time step *t*, is referred to as a decoder. Every cell of the decoder receives an input from the preceding cell, produces its very own output, and passes it to the following cell.

Within the question-answering hassle, the output sequence is a set of all words from the solution. Each word is written as y_t with t indicating the word's order. The formula (2) is used to calculate any hidden state h_t:

$$h_t = f(W^{(hh)} h_{t-1}) \quad (2)$$

As you can see in (2), we're simply computing the next concealed state by using the prior one. The formula is used to calculate the output y t at time step t:

$$y_t = softmax(W^S h_t) \quad (3)$$

We compute the outputs by combining the hidden state at the current time step with the weight W^S. SoftMax is used to generate a probability vector that will aid in the prediction of the ultimate result (e.g., word in the question-answering problem).This model's strength comes in its ability to map sequences of varying lengths to one another. The inputs and outputs, as you can see, are not correlated, and their lengths can differ. This brings up a whole new set of challenges for which such design can now be used.

E. Rearrangement

Post translation, the translated words need to be rearranged keeping the grammatical rules of the concerned language in mind. The rearrangement is important so that the translated sentence makes sense.This is a very important step in the complete translation process because even if the machine is trained with 100 percent accuracy and the word translations carried out without any errors, the output will not be ideal if the rearrangement step isn't carried out perfectly.

Example:how are you ?→ "how", "are", "you" → "कैसे", " हैं", "आप" → कैसेहैंआप". My name is May. → "My", "name", "is", "May" → "मेरा ", "नाम ", "हैं", "में"→मेरानाममेंहैं।

F. Training the dataset:

To train the machine to be able to do efficient translation we need to first train it with a diverse dataset so as to make sure that it is able to understand the grammatical nuances of both the concerned languages, "English" &"Hindi". Now we firstly needed to make sure that the dataset on which we were going to train our model is free from any errors. So, we took our dataset from Kaggle. One file was a set of English sentences, and another file was the English sentences in the previous file translated to Hindi.

Below is shown a glimpse of the English dataset.

```
new jersey is sometimes quiet during autumn , and it is snowy in april .
the united states is usually chilly during july , and it is usually freezing in november .
california is usually quiet during march , and it is usually hot in june .
the united states is sometimes mild during june , and it is cold in september .
your least liked fruit is the grape , but my least liked is the apple .
his favorite fruit is the orange , but my favorite is the grape .
paris is relaxing during december , but it is usually chilly in july .
new jersey is busy during spring , and it is never hot in march .
our least liked fruit is the lemon , but my least liked is the grape .
the united states is sometimes busy during january , and it is sometimes warm in november .
the lime is her least liked fruit , but the banana is my least liked .
he saw a old yellow truck .
india is rainy during june , and it is sometimes warm in november .
that cat was my most loved animal .
```

Fig. 2 - Sample English Dataset

And similarly, below is shown a glimpse of the Hindi dataset.

```
नई जर्सी कभी-कभी शरद ऋतु के दौरान शांत होती है, और यह अप्रैल में बर्फीली होती है।
संयुक्त राज्य आमतौर पर जुलाई के दौरान मिर्च होता है, और यह आमतौर पर नवम्बर में जम जाता है।
कैलिफ़ोर्निया आमतौर पर मार्च के दौरान शांत होता है, और आमतौर पर जून में गर्म होता है।
संयुक्त राज्य कभी-कभी जून के दौरान हल्के होते हैं, और यह सितंबर में ठंडा होता है।
आपका सबसे कम पसंद किया जाने वाला फल अंगूर है, लेकिन मेरा सबसे कम पसंद किया जाने वाला सेब है।
उसका पसंदीदा फल नारंगी है, लेकिन मेरा पसंदीदा अंगूर है।
पेरिस के दौरान पेरिस आराम कर रहा है, लेकिन यह आमतौर पर जूली में मिर्च है।
नई जर्सी वसंत के दौरान व्यस्त है, और यह मार्च में कभी गर्म नहीं होती है।
हमारा सबसे कम पसंद किया जाने वाला फल नींबू है, लेकिन मेरी सबसे कम पसंद अंगूर है।
एकजुट राज्य कभी-कभी जीवन के दौरान व्यस्त होते हैं, और यह कभी-कभी नवम्बर में गर्म होता है।
चूना उसका सबसे कम पसंद किया जाने वाला फल है, लेकिन केला मुझे सबसे कम पसंद है।
उसने एक पुराना पीला ट्रक देखा।
जून जून के दौरान बरसात होती है, और यह कभी-कभी नवम्बर में गर्म होती है।
वह बिल्ली मेरा सबसे प्रिय जानवर था।
```

Fig. 3 - Sample Hindi Dataset

Now next we needed to choose a platform on which we would train our model on. We chose to train it on Google Colab. It took us around 3 to 4 hours for the model to get trained once. The machine started the training with the loss percentage greater than the accuracy percentage. But with every epoch we were able to see an increase in the accuracy percentage and simultaneously a downfall in the loss percentage. After around the passage of 80 mins the accuracy percentage took over the loss percentage and after that the model started to increase its accuracy level exponentially. Below is the graphical representation of the validation loss and training loss of our model after the training concluded.

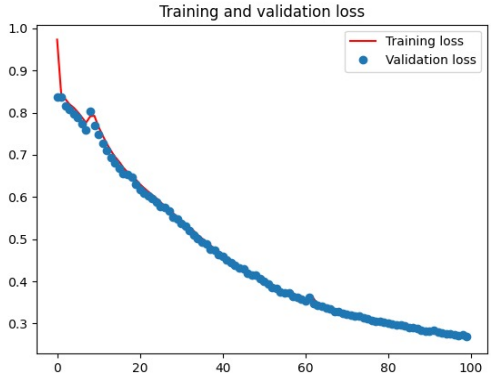

Fig. 4 - Training vs Validation Loss

Also below is the graphical representation of the training validation accuracy of our model post training.

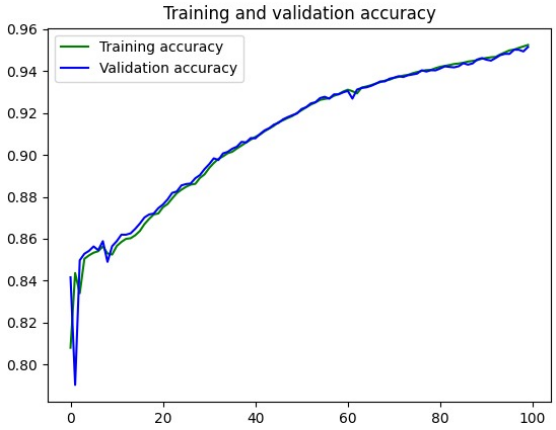

Fig. 5 - Training vs Validation Accuracy

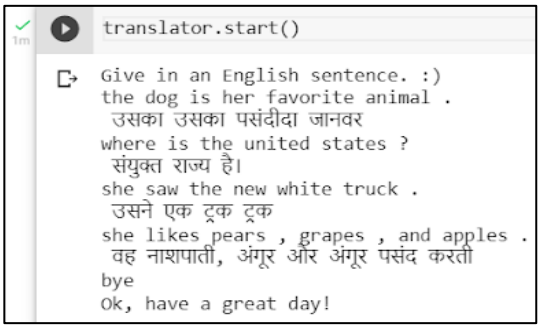

Fig. 6 - Sample Translation from English to Hindi

IV. Result and Discussion

The Sequence-to-Sequence model was used to train our proposed model, which was designed using the neural machine approach. An example of our output is shown in Figure 6. In this work, using the encoder-decoder approach, we created an English to Hindi translation model. The Kaggle English to Hindi Neural Machine translation dataset was used to train our model.

V. Conclusion

To conclude we can see that we have translated a given sentence inputted in Hindi to English and displayed the output successfully. We built a Machine translation model using a neural network and Sequence-to-Sequence model.

There have been some translations which are out of place and hence leading to a bit of temperamentin the meaning of the sentence. But with further improvement and training of the model with more diverse datasets the original meaning of the inputted sentence will be kept intact.

To do so the main issue will be to train the machine to identify proper nouns and refrain from translating them and keeping them intact during the translation process. To add to that, the grammatical rules of languages differ from each other. The differences can range from simple things to very complex things. For example, in English the word “You” is used to address someone irrespective of their age. But when we speak the Hindi language, words such as “Tu”, “Tum” and “Aap” are used for addressing someone depending on their age with respect to us. So, the word “You” can be translated to any of these three words depending on whom the sentence is being addressed to. To get such intricate details correct, a high level of training is required. Also, for future work, adding more languages to the machine can be considered. For that we need to add another part before the translation part kicks in. The machine would have to accurately detect which language is being translated and to which language the machine needs to be translated. This would be an important step as the whole process would be dependent on this step.

VI. References

[1] Sindhu C., Rajkakati D., Shelukar C. (2021) Context-Based Sentiment Analysis on Amazon Product Customer Feedback Data. In: Hemanth D., Vadivu G., Sangeetha M., Balas V. (eds) Artificial Intelligence Techniques for Advanced Computing Applications. Lecture Notes in Networks and Systems, vol 130.

[2] [2] KethanPabbi, C. Sindhu, Isukapalli Sainath Reddy and Bhumireddy Naga Sai Abhijit, “The Use of Transformer Model in Opinion Summarisation”, Webology, Vol 18, pp:1084-1095, 2021.

[3] [3] K.Chen, "A Neural Approach to Source Dependence Based Context Model for Statistical Machine Translation" IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 2, pp. 266-280, Feb. 2018, doi: 10.1109/TASLP.2017.2772846.

[4] [4] S. Mathur, V. P. Saxena, "Hybrid approach to English-Hindi name entity transliteration" 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, pp. 1-5, doi: 10.1109/SCEECS.2014.6804467, 2014

[5] [5] E. Katta and A. Arora, "An improved approach to English-Hindi based Cross Language Information Retrieval system," 2015 Eighth International Conference on Contemporary Computing, pp. 354-359, 2015, doi: 10.1109/IC3.2015.7346706.

[6] M. Bansal and G. Jain, "Improvement of English-Hindi machine translation using ConceptNet," 2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE),pp.198202,doi:10.1109/RDCAPE.2017.8358266.

[7] J. Nair, K. A. Krishnan, R. Deetha, "An efficient English to Hindi machine translation system using hybrid mechanism" 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2109-2113, doi: 10.1109/ICACCI.2016.7732363, 2016.

[8] K. Chen, "Towards More Diverse Input Representation for Neural Machine Translation" in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.28, pp.1586-1597, doi: 10.1109/TASLP.2020.2996077.

[9] C. Duan, "Modeling Future Cost for Neural Machine Translation" in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, doi: 10.1109/TASLP.2020.3042006, pp. 770-781, 2021.

[10] W. Xiong, Y. Jin, "A new Chinese-English machine translation method based on rule for claims sentence of Chinese patent," 2011 7th International Conference on Natural Language Processing and Knowledge Engineering, pp. 378-381, doi: 10.1109/NLPKE.2011.6138228, 2011.

[11] H. Sun, R. Wang, K. Chen, M. Utiyama, E. Sumita, T. Zhao, "Unsupervised Neural Machine Translation with Cross-Lingual Language Representation Agreement" in IEEE/ACM

Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1170-1182, 2020.

[12] N. Jayanthi, A. Lakshmi, C. S. K. Raju, B. Swathi, "Dual Translation of International and Indian Regional Language using Recent Machine Translation" 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pp. 682-686, doi: 10.1109/ICISS49785.2020.9316016.

[13] G. Mahmut, M. Nijat, R. Memet and A. Hamdulla, "Exploration of Chinese-Uyghur neural machine translation," 2017 International Conference on Asian Language Processing (IALP), 2017, pp. 176-179, doi: 10.1109/IALP.2017.8300573.

[14] Choudhary, M. Singh, "GB theory-based Hindi to english translation system" 2009 2nd IEEE International Conference on Computer Science and Information Technology, doi: 10.1109/ICCSIT.2009.5234543, pp. 293-297.

[15] N. Jayanthi, A. Lakshmi, C. S. K. Raju, B. Swathi, "Dual Translation of International and Indian Regional Language using Recent Machine Translation" 3rd International Conference on Intelligent Sustainable Systems, pp. 682-686, doi: 10.1109/ICISS49785.2020.9316016,2020.

[16] K. SaengthongPattana, K. Kriengket, P. Porkaew and T. Supnithi, "Thai-English and English-Thai Translation Performance of Transformer Machine Translation" 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), pp.1-5, doi: 10.1109/iSAI-NLP48611.2019.9045174, 2019.

[17] K. Chen, R. Wang, M. Utiyama, E. Sumita and T. Zhao, "Neural Machine Translation with Sentence-Level Topic Context" in IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp. 1970-1984, Dec. 2019, doi: 10.1109/TASLP.2019.2937190, vol. 27, no. 12,

[18] H. Sun, R. Wang, K. Chen, M. Utiyama, E. Sumita, T. Zhao, "Unsupervised Neural Machine Translation with Cross-Lingual Language Representation Agreement" in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1170-1182, 2020.

[19] X. Wang, Z. Tu and M. Zhang, "Incorporating Statistical Machine Translation Word Knowledge into Neural Machine Translation," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 12, pp. 2255-2266, Dec. 2018, doi: 10.1109/TASLP.2018.2860287.