Machine Translation between Tamil and English using Transformer Model's Positional Encoders.

[1]K Shanmukapriya
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
shanmukapriya.k2020@vitstudent.ac.in

[2]G Nikhil Vaidhyanathan
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
nikhilvaidhyanathan.g2020@vitstudent.ac.in

***Abstract*— The field of Natural Language Processing (NLP) has seen tremendous growth, and machine translation is one of its most significant applications. Translation between two languages has become a crucial need for communication and business transactions in a globalized world. However, manual translation is often time-consuming, expensive, and prone to errors. This has led to the development of automated machine translation systems, where Artificial Intelligence (AI) plays a critical role.**

The Transformer model is a state-of-the-art neural network architecture that consists of an encoder-decoder structure for machine translation. In this study, the input text is preprocessed by tokenizing and encoding the text into vectors, and the positional encoder maintains the positional information of the input text, which is critical for accurate translation.

The model is trained using parallel corpus data, which contains pairs of sentences in Tamil and English. The study investigates the impact of different hyper-parameters on the model's performance, such as the number of layers in the encoder and decoder, the size of the hidden layer, and the learning rate. The findings suggest that increasing the number of layers and the size of the hidden layer can improve the model's performance.

The limitations of the model are discussed, such as the need for more datapoints and more computational power while using a parallel corpus and the need for unsupervised machine translation techniques to train the model without parallel corpus data. In conclusion, this study demonstrates the effectiveness of positional encoder in the Transformer model for Tamil to English and English to Tamil translation, and the potential of AI for improving machine translation.

Keywords*— *natural language processing (NLP), artificial intelligence (AI), transformer model, encoder-decoder, Tamil to English translation, English to Tamil translation, parallel corpus data.

I. INTRODUCTION

In recent years, Artificial Intelligence (AI) has been a major game-changer in various fields, including natural language processing (NLP). With the advancement of AI, machines have become capable of understanding and processing human language, leading to the development of machine translation systems. These systems have the ability to translate text from one language to another, and this has become a key focus area in the field of NLP. With the growing need for language translation in today's globalized world, machine translation systems have become increasingly important.

One of the areas where machine translation systems are being used extensively is in the translation of Tamil to English and English to Tamil. Tamil is one of the oldest languages in the world, spoken by over 70 million people globally. Due to its cultural richness, Tamil literature, films, and music have a vast following across the world. With the growing popularity of Tamil, there is a growing need for accurate and efficient translation systems that can translate Tamil text into English and vice versa. This is where AI and NLP come into the picture, with the development of machine translation systems that are specifically designed for Tamil to English and English to Tamil translation.

This paper presents a study on Tamil to English and English to Tamil translation using AI concepts and positional encoder in Transformer model. The Transformer model is a state-of-the-art neural network architecture for machine translation and has been successful in producing high-quality translations. The model consists of an encoder-decoder architecture, where the encoder processes the input text, and the decoder generates the output text.

The input text is preprocessed by tokenizing and encoding the text into a sequence of vectors, which can be fed into the Transformer model. The positional encoder in the Transformer model allows the model to maintain the positional information of the input text, which is crucial for accurate translation. The model is trained using parallel corpus data, which consists of pairs of sentences in Tamil and English.

The study explores the effect of various hyper-parameters on the model's performance, including the number of layers in the encoder and decoder, the size of the hidden layer, and the learning rate. It has been found that increasing the number of layers and the size of the hidden layer can lead to better results.

However, the model has limitations, including the requirement for more data points and computational power while using a parallel corpus. Furthermore, the model can be improved by implementing unsupervised machine translation techniques has been recognized to train the model without parallel corpus data. Despite these limitations, the study demonstrates the effectiveness of positional encoder in the Transformer model for Tamil to English and English to Tamil translation, and the potential of AI for improving machine translation. Future research should be conducted to develop a more robust and efficient model that can overcome the limitations of the existing model.

II. LITERATURE SURVEY

Machine translation systems have become a popular research area in recent years due to their potential to break down language barriers and increase intercultural dialogue. In this review of the literature, we will look at few studies that suggest machine translation models for Tamil to English and English to Tamil translation.

Sivakumaran (2021) emphasised the challenges of translating books from English to Tamil due to the language's intricate sentence structure, vocabulary, and syntax. According to the study, it is difficult for machine translation algorithms to maintain the original text's tone and meaning while translating between these two languages.[1]

Using a rule-based methodology, Renganathan (2002) created an interactive web-based machine translation system for Tamil to English. The Tamil sentence was generated by applying a set of rules after the English sentence's syntactic structure was determined by the system using an English parser. The study's results were encouraging, but it also made notice of the inadequacies of rule-based methods for dealing with the complexity of natural language.[2]

A sequence-to-sequence model was used by Jain et al. (2020) to develop a neural machine translation system for translating Tamil into English. With a short dataset, the study's bidirectional LSTM network with attention mechanism produced successful results. The size of the training dataset and the sort of attention mechanism were two other variables that the study examined in relation to their effects on translation quality.[3]

A Tamil character recognition, translation, and transliteration system was created by Prakash et al. (2020) that can read Tamil handwriting, translate it into English, and then transliterate it into the Roman script. High character recognition and transliteration accuracy were attained using a deep learning model with convolutional and recurrent layers. [4]

An English to Tamil neural machine translation system utilising a transformer model was proposed by Choudhary et al. (2018). The study included multiple strategies, including back-translation and the development of synthetic data, to enhance the translation quality in addition to a pre-trained model with fine-tuning on the target language. In the Third Conference on Machine Translation's joint task, the study produced results that were competitive.[5]

Using OCR and NLP methods, Manigandan et al. (2017) created a Tamil character recognition system that can identify characters from old Tamil inscriptions. Character recognition in the study was highly accurate thanks to a combination of feature extraction, segmentation, and classification methods.[6]

The complicated syntax and sentence structure of Tamil make it challenging to create machine translation systems, as Fredric Gey noted in his 2002 study "Prospects for Machine Translation of the Tamil Language." He underlined the necessity for parallel corpora, saying that they would assist machine translation systems be more accurate.[7]

Tayebeh Mosavi Miangah and M. Dehcheshmeh explored the effect of utilising parallel corpora on the quality of machine translation in "The effect of using parallel corpora on translation quality" (2012). They discovered that the use of parallel corpora enhanced the system's capacity to learn the relationship between the source and destination languages, which improved the quality of machine translation systems.[8]

In his article, S. R. (2019) suggested a machine translation system for Tamil to English. The suggested system was built to handle complicated sentences and followed a rule-based methodology. The system's performance was assessed using the BLEU score, and the findings indicated that the translation quality had improved. [9]

B.N. Narasimha Raju and M.S. Bhadri Raju created a statistical machine translation system for Indian languages in "Statistical Machine Translation System for Indian Languages" (2016). The suggested system generated translations using the phrase-based model, a language model, and a translation model. The proposed system, according to the authors, results in better translation quality for Tamil and other Indian languages.[10]

III. POSITIONAL ENCODING IN TRANSFORMER MODEL

A. What is transformer model's positional encoding?

The Transformer model is a neural network architecture that has revolutionized natural language processing tasks such as machine translation, language modeling, and text generation. The Transformer model is based on the attention mechanism, which enables it to capture the context of the input sequence efficiently. However, unlike traditional recurrent neural networks, the Transformer model does not use recurrence to process sequential data. Instead, it relies on the self-attention mechanism to capture the relationship between different elements of the input sequence. In this section, we will explain the concept of positional encoding in the Transformer model, which enables it to process the input sequence in a non-sequential manner.

The Transformer model uses an input embedding layer to represent the input sequence as a fixed-size vector. The input embedding layer maps each word in the input sequence to a high-dimensional vector space, which captures the semantic and syntactic information of the word. However, the input embedding layer does not capture the position of the word in the input sequence, which is crucial for capturing the context of the sequence. Therefore, the Transformer model uses a positional encoding technique to encode the position of the words in the input sequence.

The positional encoding is added to the input embedding vector for each position in the sequence. The positional encoding vector is computed using a mathematical function that encodes the position of the word in the input sequence. The function is designed to generate unique encoding vectors for each position in the sequence, which enables the Transformer model to capture the context of the sequence efficiently.

The mathematical function used to compute the positional encoding vector is based on sine and cosine functions. The positional encoding function is defined as follows(1):

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

(1)

As an example, let's consider the sentence "I love cats". In this case, the first token "I" will have a positional encoding of (2):

$$PE_{(1,1)} = \sin(\frac{1}{10000^{2\cdot 0/512}}) = \sin(0) = 0$$

(2)

And the second token "love" will have a positional encoding of (3):

$$PE_{(2,1)} = \sin(\frac{1}{10000^{2\cdot 1/512}}) = \sin(0.0061) \approx 0.0061$$

(3)

Similarly, the third token "cats" will have a positional encoding of (4):

$$PE_{(3,1)} = \sin(\frac{1}{10000^{2\cdot 2/512}}) = \sin(0.0244) \approx 0.0244$$

(4)

Thus, each token in the sentence will have a unique positional encoding that reflects its position in the sequence. To summarize, the positional encoding in the Transformer model is used to add positional information to the input embeddings so that the model can distinguish between tokens based on their position in the sequence. This is done using a combination of sine and cosine functions with varying frequencies and offsets, resulting in a unique encoding for each position in the sequence. The positional encoding is added to the input embeddings and is passed as input to the encoder and decoder layers in the Transformer model.

B. *Why transformer model's positional encoding used in machine translation systems?*

Positional encoding is a critical component in the Transformer model, which is widely used for machine translation tasks. This technique helps to embed the positional information of words into the input sequences, which is essential for accurately translating a sentence from one language to another. Here are some ways in which positional encoding benefits machine translation systems:

1) *Overcoming Word Order Ambiguity*

• Natural languages use the order of words in a sentence to convey meaning.

• The Transformer model uses positional encoding to embed the position of each word in the sentence, helping the model learn the relative positions of words and their importance in the sentence.

• This improves the accuracy of translations.

2) *Handling Long Sentences*

• Machine translation systems may struggle with long sentences.

• The Transformer model addresses this by using self-attention mechanisms that can attend to all words in the input sequence simultaneously.

• Positional encoding is crucial to help the model differentiate between words and their positions in long sequences.

3) *Capturing Context-Dependent Translation*

• The Transformer model uses an encoder-decoder architecture to translate input sentences.

• Positional encoding helps to capture the context of the input sentence, allowing the decoder to produce a more accurate translation.

• Positional encoding enables the model to understand context-dependent translations, such as the different meanings of the word "like" in the sentence "I like apples, but I don't like bananas."

4) *Transfer Learning*

• Transfer learning involves using pre-trained models for a specific task and then fine-tuning them for a related task.

• Positional encoding is crucial for transfer learning in machine translation as it helps to transfer positional information from the source language to the target language.

• The Transformer model can use the same positional encoding technique for both the source and target languages to learn to translate sentences accurately.

In conclusion, the positional encoding technique in the Transformer model is an essential component of machine translation systems. It helps to embed the positional information of words into input sequences, thereby addressing challenges such as word order ambiguity, handling long sentences, capturing context-dependent translations, and enabling transfer learning. By using positional encoding, machine translation systems can achieve higher accuracy and better performance, thereby making it possible to provide accurate translations across different languages.

IV. PROPOSED METHODOLOGY

We used the concepts of transformer model's positional encoding and the knowledge we learnt from our literature survey to implement Tamil to English and English to Tamil translation. We have elaborated the methodology we used to how the machine is trained to translate the texts in simple words below:

a. Clean up the data : We convert all the data to lowercase, remove any unnecessary spaces and remove special characters.

b. Build the vocabulary: We build the vocabulary using transformer model's positional encoders by vector indexing them.

c. Padding Data: We are padding data since the words are of different lengths.

d. Train and Test:

• Using transformers we train the machine in a self-supervised fashion.(Self-supervised learning is a training method in which the model's inputs are used to autonomously calculate the objective.)

• Here we use 200 datapoints for Tamil to English translation where 80% of the data is used for training the model and 20% of the data is used for validating the model.

• Similarly for English to Tamil translation we use 1000 datapoints and the same 80-20% ratio for training and validating the model.

e. Saving the model.

We use a parallel corpus data because machine translation systems often use them and it can help in achieving high quality output. Depending on the hardware configuration of the computer or laptop we used, the number of datapoints were picked out accordingly.

The number of encoding and decoding layers used is 6. We used 150 epochs for Tamil to English translation and 70 epochs for English to Tamil translation.

The methodology is given a flowchart below (Figure-1) .:

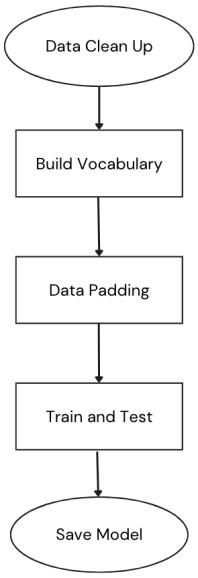

Figure-1.

V. RESULT AND DISCUSSION

The following results were obtained after implementing the proposed methodology:

A) *Tamil to English Translation:*

The time taken to finish all epochs and train the model was approximately 2 hours.

We used the following sentences to check if the model produced the desired results (The actual meaning of the sentence is also given below the sentence.):

• இந்த ஆப்பிள் இனிப்பாக இருக்கிறது

"This apple is sweet"

• நான் பள்ளிக்கு நடந்து செல்கிறேன்

"I walk to school"

• என் மகனைப் பற்றி பெருமைப் படுகிறேன்

"I'm proud of my son"

The table below elaborates how the sentence is translated word by word (Table-1, 2, 3):

Tamil	English
என்	im
மகனைப்	proud
பற்றி	of
பெருமைப்	my
படுகிறேன்	son

Table - 1

Tamil	English
நான்	i
பள்ளிக்கு	school
நடந்து	school
செல்கிறேன்	to school

Table - 2

Tamil	English
இந்த	this
ஆப்பிள்	apple
இனிப்பாக	is
இருக்கிறது	sweet

Table - 3

From the above tables we can observe that most of the sentences are translated correctly but except for the 2nd sentence (Table-2). Here the model failed to translate the sentence to "I walk to school" instead it translated to "I school school to school", it is almost correct but it missed few vocabulary.

But this might be due to the reason of limited number of datapoints we used to train the model, since we had very less computational power.

For evaluation purpose we have plotted training loss and validation loss in a graph (Figure-2):

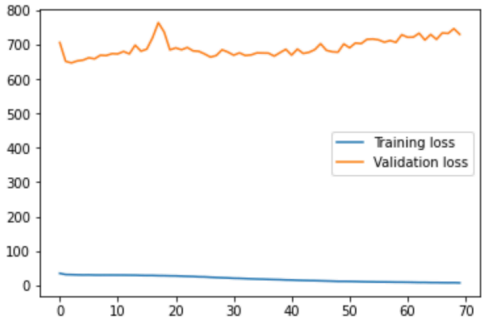

Figure-2

B) *English to Tamil Translation:*

The time taken to finish all epochs and train the model was approximately 8 hours.

We used the following sentences to check if the model produced the desired results (The actual meaning of the sentence is also given below the sentence.):

- I want to become an astronaut in the future

"நான் எதிர் காலத்தில் விண்வெளி வீரராக மாற விரும்புகிறேன்"

- I'm kind of happy

"நான் ஒரு விதமான மகிழ்ச்சியிலிருக்கிறேன்!"

- She gave him a book

" அ வ ள் அ வ னு க்கு ஒ ரு பு த் த க த் தை க் கொடுத்தாள்"

The table below elaborates how the sentence is translated word by word (Table-4, 5, 6):

English	Tamil
I	நான்
want to	எதிர்
become	காலத்தில்
an astronaut	ஒரு
in	விமானியாக
the future	விரும்புகிறேன்

Table - 4

English	Tamil
I'm	நான்
kind	ஒரு
of	விதமான
happy	மகிழ்ச்சியிலிருக்கிறேன்

Table-5

English	Tamil
She	அவள்
gave	அவனுக்கு
him	ஒரு
a	புத்தகத்தைக்
book	கொடுத்தாள்

Table-6

From the above tables we can observe that most of the sentences are translated correctly but except for the 1st sentence (Table-4). Here the model failed to translate the sentence to "நான் எதிர் காலத்தில் விண்வெளி வீரராக மாற விரும்புகிறேன்" instead it translated to " நா ன் எ தி ர் கா ல த்தி ல் ஒ ரு வி மா னி யா க விரும்புகிறேன்", which actually means "I want to become a pilot in the future".

The model has tried to translate the word astronaut to the nearest similar word. Since we used more data points than Tamil to English translation the model learns the position and finds similar sentences and words to translate as it is trained better and has more intelligence. The model had also flagged astronaut is not in its vocabulary, in the output(Figure-3).

"`astronaut is not in vocab`"

Figure-3.

For evaluation purpose we have plotted training loss and validation loss in a graph (Figure-4).

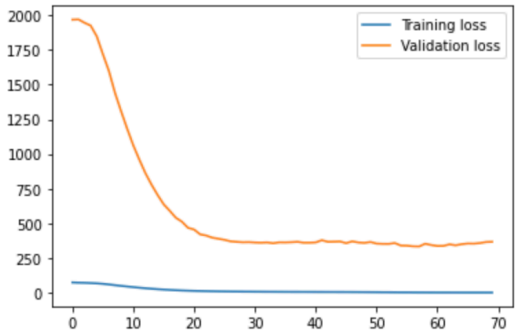

Figure-4

From observing the results from both of the translations, we understand that with more datapoints and more computational power we can achieve great results. But since we used a free Jupyter notebook environment that runs entirely in the cloud our resources were very limited. NLP in general requires extensive computational power to produce good results.

But one major drawback is that the subject-predicate logic of Tamil and English is very different.

Many English words can mean the same word in Tamil but the subject-predicate logic comes to a big play in Tamil, since the context and tone can change the sentence structure and meaning with just a different placement of the word. Moreover we can easily map many words to one but it's hard to map one word to many. Let's understand from an example(Figure - 5) :

"கலை" - Art

Stag (Male deer)

Cloth

Saddle of horse

Figure-5.

Hence to map one word to many words will require more datapoints and more computational power.

We also found in our literature survey that by using unsupervised machine translation techniques and models, we can eliminate the need for using a parallel corpus. We generate faster and accurate results without using a parallel corpus. We can train the machine to perform two tasks simultaneously: translation and reconstruction of the original input.

The machine is able to learn a shared representation of the input and output languages as

a result, making it possible for it to carry out precise translations even for languages with limited parallel data. One example of unsupervised machine translation model is Neural Machine Translation model.

VI. Conclusion

A Tamil to English and English to Tamil translation models are created successfully with limited data and limited resources. The model can produce results effectively but with more datapoints and more computational power the model can produce better results. Also by eliminating the dependency of parallel corpus the model can learn and understand the position words better, this will enable us in training the model better.

For further work and improvisation of the model we can use unsupervised machine translation techniques and models instead of transformer models for faster and accurate results.

VII. References

(1) Sivakumaran, Mathura. (2021). Difficulties in Translating Texts from English to Tamil.

(2) Renganathan, V. (2002). An interactive approach to development of english-tamil machine translation system on the web. In The international Tamil Internet 2002 Conference and Exhibition (TI2002).

(3) Jain, M., Punia, R., & Hooda, I. (2020). Neural machine translation for Tamil to English. Journal of Statistics and Management Systems, 23:7, 1251-1264, DOI: 10.1080/09720510.2020.1799582

(4) M. Prakash,, Apoorva Ojha, & Priyanshu Raman. (2020). Tamil Character Recognition, Translation and Transliteration System. International Journal of Engineering and Advanced Technology (IJEAT), 9(4), 1762–1767. https://doi.org/10.35940/ijeat.D7633.049420

(5) Choudhary, H., Pathak, A. K., Saha, R. R., & Kumaraguru, P. (2018, October). Neural machine translation for English-Tamil. In Proceedings of the third conference on machine translation: shared task papers (pp. 770-775).

(6) Manigandan, T. V. V. D. V. N. B., Vidhya, V., Dhanalakshmi, V., & Nirmala, B. (2017, August). Tamil character recognition from ancient epigraphical inscription using OCR and NLP. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 1008-1011). IEEE.

(7) Gey, Fredric. (2002). Prospects for Machine Translation of the Tamil Language. Tamil Internet.

(8) Mosavi Miangah, Tayebeh & Dehcheshmeh, M.. (2012). The effect of using parallel corpora on translation quality. Translation Studies. 97-112..

(9) S, R. (2019). ENGLISH TO TAMIL MACHINE TRANSLATION SYSTEM. Language in India Www.languageinindia.com ISSN 1930-2940 Vol. 19:5 .

(10) Narasimha Raju, B.N., & Bhadri Raju, M.S. (2016). Statistical Machine Translation System for Indian Languages. 2016 IEEE 6th International Conference on Advanced Computing (IACC), 174-177.

(11) https://huggingface.co/course/chapter1/4#:~:text=Transformers%20are%20language%20models&text=This%20means%20they%20have%20been,needed%20to%20label%20the%20data!

(12) https://towardsdatascience.com/master-positional-encoding-part-ii-1cfc4d3e7375

(13) https://towardsdatascience.com/master-positional-encoding-part-i-63c05d90a0c3

(14) https://kikaben.com/transformers-positional-encoding/

(15) https://agarathi.com/word/%E0%AE%95%E0%AE%B2%E0%AF%88

(16) https://medium.datadriveninvestor.com/transformer-break-down-positional-encoding-c8d1bbbf79a8

(17) https://engineering.fb.com/2018/08/31/ai-research/unsupervised-machine-translation-a-novel-approach-to-provide-fast-accurate-translations-for-more-languages/

(18) https://paperswithcode.com/task/unsupervised-machine-translation

(19) https://gengo.com/translation-corpus/#:~:text=A%20parallel%20text%20translation%20corpus,to%20achieve%20high%2Dquality%20output.

(20) https://mt.cs.upc.edu/2021/03/08/major-breakthroughs-in-unsupervised-neural-machine-translation-v/