International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

Web Compatible Language Translator Using Deep Learning Model

Divakar V
Assistant Professor
Mahin Kumaar T N
Computer Science and Engineering,
Mathan S
Computer Science and Engineering

Computer Technology Bannari Amman Institute of Technology Bannari Amman Institute ofTechnology Bannari Amman Institute of Technology

Kishor Kumar K T
Computer Science and Engineering
Bannari Amman Institute of Technology

Abstract

Deep learning models have recently demonstrated enormous advancements in machine translation. These systems require a lot of memory and have high processing costs. This paper offers an overview of deep learning architectures used in machine translation. This paper focuses on designing a transformer-based language translator and analyzes the per formance of language translators by comparing other models. Despite the large number of studies that have been offered over the past few years, nothing has been done to examine how this new technological transformer is developing.This analysis of the existing literature's traces the origins of the ideas behind neural machine translation, examines the key branches, and contends on some possible future research directions in this field.

Index Terms—deep learning, machine translation, NLP, RNN, transformer

I. Introduction

The study of how computers interpret and understand human languages falls under the field of natural language processing (NLP). The area has relied more and more on data-driven com putation involving statistics, probability, and machine learning since the 1980s. Artificial neural networks (ANNs), some times with billions of trainable parameters, are used in ”deep learning,” which is made possible by recent improvements in computer power and parallelization, harnessed by graphical processing units (GPUs). Additionally, the training of such deep architectures is made possible by the modern availability of massive data sets, made possible by sophisticated data collection procedures.

The traditional NLP sub-field of machine translation (MT) studies how to utilize computer software to translate text or speech from one language to another without the need for a human translator. Since the ultimate goal of NLP and AI is to fully understand human text (voice) at the semantic level, the MT problem has drawn a lot of interest in recent years. In addition to its scientific significance, MT offers enormous potential to reduce labor costs in many real-world contexts, including academic communication and international commercial negotiation. There has been a lot of studies done on machine translation, and in recent years, numerous effective methods have been proposed. Recent advancements in Deep Learning have led to the emergence of a novel technique known as Neural Machine Translation (NMT). NMT has a great deal of potential to become a new trend in society.

Following the development of a simple model, numerous NMT models have been put forward, some of which have made significant advancements with cutting-edge outcomes. This paper examines the future direction of NMT while outlining its main branches and most recent advancements.

Ii. Literature Survey

Despite the lack of much research on NMT, several other studies are closely connected.It have described the traditional techniques for learning sequences. [1] this gave crucial details about the history of NMT and the underlying knowledge [3]. They have done some model comparison work in NMT with experiments and evaluation of the practical performance of some widely accepted technologies. then they have rare theoretical analysis, particularly in displaying the relationship between many proposed models [4]. Some researchers, on the other hand, focused their survey work on a specific aspect of NMT, or the Attention Mechanism, [6] although both of them had a broad reach that was directed to all different sorts of AI activities using Attention. However, the direct and recent literature survey on NMT is the main topic of this research. We have extensively researched the pertinent literature in this new trend and offered a detailed analysis for the current standard technology in NMT.

Iii. Deep Learning Based On Machine Translation

Text translation from one language to another is the main goal of the computational linguistics discipline and it is referred to as machine translation (MT). Due to the capabilities of deep learning, Neural Machine Translation (NMT) has become the most effective algorithm for performing this task. Tech businesses all across the world are investing heavily in NMT, even though Google Translate is the main industry example. This cutting-edge approach uses deep learning to develop a model that can translate between any two languages by using enormous datasets of translated sentences. Numerous NMT variants are currently being researched and used in the industry as a result of the extensive

sentence to the target sentence. In fact, both languages can use this intermediate vector to describe the same semantic meaning.The network's model selection and RNN-based NMT models diverge in two major ways. (a) the directionality; (b) the type of activation function. Directionality: Bidirectional RNN shows more performance by comparing unidirectional research conducted in recent years.

A. Recurrent Neural Networks and Long Short-Term Memory Networks:

Recurrent neural network (RNN) is a widely used kind of recursive neural network. It helps to remember the previous items when processing new ones because a lot of natural language processing (NLP) depends on the sequence of words or other elements, such as phonemes or sentences. There are occasionally backward dependencies, meaning that the proper processing of some words may depend on terms that come after them. As a result, it is advantageous to use two RNN layers and combine their outputs to analyze words in both forward and backward orientations [2]. A bidirectional RNN is this configuration of RNNs. A series of RNN layers may also result in a superior final representation. This might allow input effects to last longer than with a single RNN layer, enabling longer-lasting effects. An RNN stack is this arrangement of RNN cells in sequential order. The long short-term memory (LSTM) network is one sophisticated RNN. The recursive nodes in LSTMs are made up of a number of distinct neurons coupled in a way that is intended to remember, forget, or reveal particular information. Although generic RNNs with a single neuron feeding back on themselves are technically capable of remembering past results, these findings become less accurate with each subsequent iteration. Sometimes, it’s crucial to recall details from the distant past, while other information from the present might not be as crucial. This essential information can be stored for a lot longer by utilizing LSTM blocks, whereas unimportant information can be forgotten. The gated recurrent unit (GRU), a little less complex LSTM version, has been demonstrated to perform as well as or better than normal LSTMs in several applications.

B. Decoder And Encoder

Encoder-Decoder’s structure is fundamentally straightfor ward. Its architecture includes two interconnected networks (the encoder and the decoder), each of which is for a particular stage of the translation process. Each hidden state of the encoder network compresses the variable-length sequence into a fixed-length vector after reading the source sentence word by-word when it gets a source sentence. This method is known as encoding. The decoder then performs the opposite task by converting the thought vector to the intended sentence word by word after receiving the encoder’s final concealed state. This procedure is also known as end-to-end translation since the encoder-decoder structure directly handles the trans lation task from the source data to the target result, meaning there is no discernible outcome in the middle process. The Encoder-Decoder structure of NMT works on the basis of a semantic space intermediate vector that maps the source RNN.Bidirectional RNN was used on the bottom layer as a substitute to capture the context information in practice [3]. The first layer of this structure reads the sentence ”left to right,” and the second layer reads it the other way. They are then joined together and given to the following layer. Word dependency between the first and last words is difficult to capture by the thought vector when using unidirectional RNN since the model has seen too many states during all time steps. Bidirectional RNNs on the country offer an extra layer of information with words read in the opposite way, which could naturally shorten this relative duration by steps. Activation Function Selection:There are three popular options for activation functions: vanilla RNN, Long Short Term Memory (LSTM), and Gated Recurrent Unit(GRU). Nonlinear activation functions are used to avoid vanishing gradient problems.

Iv. Attention Mechanisms And Transformer

NMT still experiences a significant performance hit when the original text gets longer. The primary drawback of the orig inal NMT Encoder, in comparison to other feature extractors, is that it must compress one sentence into a fixed-length vector. The network’s final output is a fixed-length vector, which may have limitations in expressing the entire phrase and result in some information loss, which causes performance to degrade as the input sentence lengthens. This led to the emergence of the Attention Mechanism. The attention Mechanism technique was initially employed as a supplement . [3] to help with the decoding process by adding additional word alignment information. This helps to reduce information loss when the input sentence is too long.

A. Structure of Attention Mechanism

The Attention Mechanism is implemented in a variety of ways. Here, we merely provide a thorough explanation of the Attention Mechanism, which is acknowledged as having made a substantial contribution to the advancement of NMT. Bahdanau et al were the ones who initially proposed the structure of the attention mechanism. Later, Luong et al.extended this work and proposed a comparable structure with minor differences. During the decoding phase, the decoder won’t predict the word; instead, it will rely only on its own data. But it works along with the attention layer to obtain the translation. The hidden states in the top layer of the encoder and the current decoder serve as the input for the

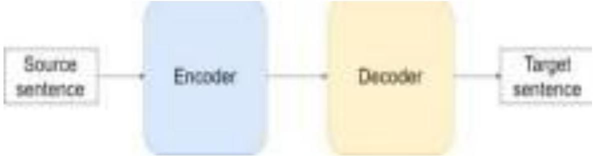


Fig. 1. The concept of Attention Mechanism, which instead of just employing information in a fixed-length of a vector can give additional alignment information.

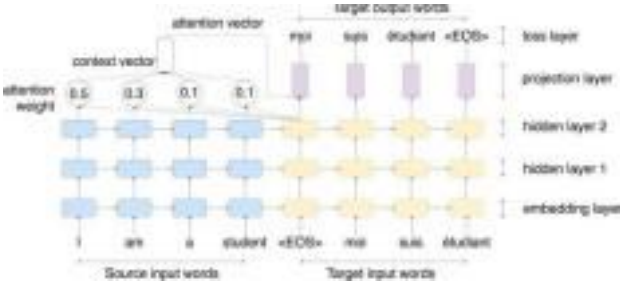


Fig. 2. The concept of Attention Mechanism, which instead of just employing information in a fixed-length of a vector can give additional alignment information.

attention mechanism. Calculating the following steps yields the relativity order:

- The attention weights score s_t will be derived by comparing the currently decoding hidden state h_t with all source states ht
- The normalization operation for all attention weight scores drives the attention weights a_t.
- Next, using the attention weights, we compute a context vector called c_t that represents the weighted average of the source states.
- To produce the final attention vector, concatenate the context vector with the current decoding hidden state. • In the following time step, the attention vector is provided as input to the decoder.

$$c_t = \sum_s a_t h_s$$ [Context vector]

The score function, among the others, could be defined in a variety of ways. Below are two traditional definitions:

$$\text{score}(h_t, h_s) = \begin{cases} h_t^T W h_s & \text{Luong's version} \\ v_a^T \tanh(W_1 h_t, h_s) & \text{Bahdanau's version} \end{cases}$$

Returning to the decoding phase, it gets data from both the attention vector and the hidden state of the decoder, and using those two vectors as a starting point, predicts words by aligning them to a new vector. It then typically has a second layer that predicts the current target word.

B. Global Attention And Local Attention

Global Attention: The technique of attention mechanism we discussed earlier is called global attention, and it is also re

$$s_t = \text{score}(h_t, h_s)$$ [Attention function]

$$a_t = \frac{\exp(s_t)}{\sum_{s=0}^{S}\exp(s_t)}$$ [Attention weight]

result in better results, this approach performs very well. A simple demonstration is shown in Fig. 3. Since one hidden state will be generated in one Encoder time-step, the main negative is that computation performance degrades when the sequence is very long. Additionally, the cost of the scoring function would increase linearly with the number of Encoder time steps.

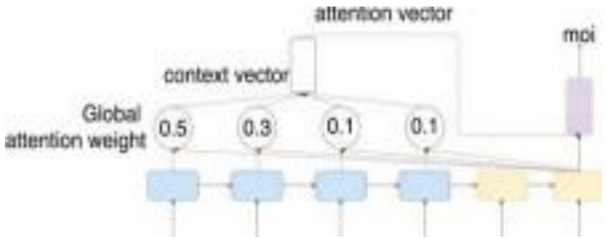


Fig. 3. Global attention and current decoder hidden states are calculated using all of the hidden states on the source side to obtain alignment data.

Local Attention: [1] It was the ones who first suggested local attention. On the other hand, as shown in Fig. 4, this model will only determine the relevance using a portion of the source text. In contrast to global attention, it fixes the length of the attention vector by supplying a scope number, avoiding the costly computation required to obtain context vectors. The study's findings showed that local attention can maintain a balance between model performance and process ing speed. Theoretically, covering more material should yield better results, but this method's outstanding result has shown that, when refined, it can get equivalent results with worldwide attention. It would seem that the current word would naturally have a high dependency on some of its neighboring terms due to a common phenomenon in human language, which is somewhat comparable to the assumption made by the n-gram language model. In the specifics of the calculation process, ferred to as a fluent kind in many attention mechanisms[1].The concept of global attention is also the basis of the attention mechanism; the correct phrase is local attention. The word ”global” comes from the context vector’s calculation, which takes into account the order in which each word in the source sentence is relevant. Since more alignment details typically

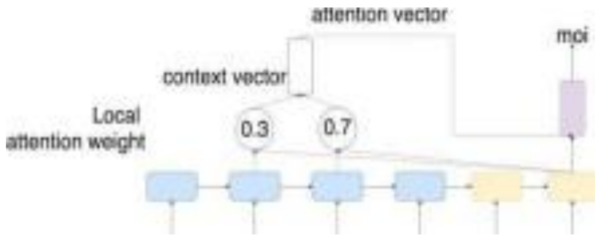


Fig. 4. The idea of local attention, which calculates the current hidden state using a sample of all the hidden states on the source side.

the model sets the context vector in scope D given the current target words location pt. Following that, the context vector $c^t$is created as a weighted average over the set of source concealed states falling between [pt D, pt + D]. Scope D is chosen based on experience, and the process is similar to that used to create the global attention vector.

C. Self-attention

Self-attention, also known as intra-attention and is used in NMT tasks as a result of the appearance of the Transformer.While other commonly observed Attention Mechanisms calculated the word dependencies between the source sequence and the target sequence to drive the context information. Self-attention determines the relationship between the words within the sequence and provides an attention-based representation of the sequence. The first three vectors that are based on the original embedding and are given to self attention are the Query vector, Key vector, and Value vector. The attention weights were then determined in the following and multi-head attention are the key structural differences in the transformer model. The transformer handles input data in a manner that is substantially distinct from recurrent or convolution models. As we previously said, the transformer uses three different types of vectors to handle the input for the computation of self-attention. They are vectors of Key, Value, and Query. And the three matrices that we trained throughout the training process are multiplied by the input embedding to drive all of these vectors. The main innovation of Transformer is self-attention. However, in practice, the scaled dot product attention is processed by the multi-head mechanism several times in parallel rather than just once, and

!

manner:

Attention(Q, K, V) =$\underline{QK_T}$

the outputs of this independent attention are then concatenated and linearly transformed into the desired dimensions. Multi

√

head Self-Attention is the name given to this multiple times

d

k

Self-Attention computation, which is used to enable the model

where the $\sqrt{}d_k$ is a scaled factor used to prevent the oper ation of dot products from creating more stable gradients. Additionally, the aforementioned computation may be used to multiply metrics, making it simple to translate the concept of dependency into related metrics.

[6] have presented a brand-new NMT con struction called the

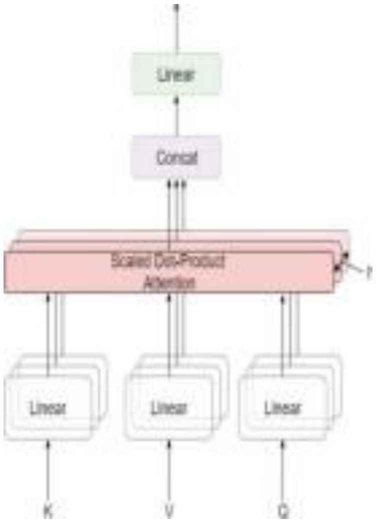


Fig. 5. The idea of local attention, which calculates the current hidden state using a sample of all the hidden states on the source side.

D. Transformer and Transformer based models Vaswani et al.

transformer. In contrast to current NMT models, it has avoidedusing the conventional RNN/CNN ar chitectures in favor of creating novel multi-layer self-attention blocks that combine apositional encoding technique. The advantages of both RNN and CNN-based models are utilized in this new trend in structure design, which has also been employed to initialize the input representation for other NLP tasks. The transformer is a comprehensive Attention-based NMT model, which is noteworthy. Transformer model has a unique structure, with the input representation and multi-head attention being the main differences. The input representation

to jointly attend to data from various representation sub-spaces at various places. The encoder is made up of six similar parts, each of which has a fully connected network above a multi head attention layer. Both layer normalization and residual connection are available in these two sublayers. The output data for each sub-layer has the same dimension, which is 512. [4] However, the decoder is more challenging. Additionally, six components are stacked, with three sublayers connected in each component, including two multi-head self-attention sublayers and one fully connected neural network sublayer. In order to prevent locations from attending to subsequent positions and to prevent the model from looking into the future of the target sequence when predicting the present word, the bottom attention layer is specifically changed using a technique termed masked. Transformer incredible perfor mance boost has drawn a lot of attention from researchers who are working on comparable refinements. The widely acknowledged shortcomings of the standard

Transformer in clude the absence of recurrence modelling, the fact that it is not theoretically Turing-complete, the inability to capture position information, and the complexity of the model. Some modifications have been suggested to fix these issues and create a better model.

V. Model Implementation And Analysis

A. Data Preparation for transformer

English to french Data is downloaded from the TensorFlow datasets. Text tokenization is performed on the text data. Text is divided into tokens during the tokenization process. These tokens may be sentence fragments, words, subwords, or characters, depending on the tokenizer. One pre-step is to convert the words to vectors, which creates a correct form that the neural network may receive, before providing the training data to the model. The most common V words in one language are often chosen, and each language has a unique word list. The pre-trained word embedding vectors like word2vec or Glove vector can also be applied directly, even if the embedding weights will be learned during the training phase. The embedding vectors receive a ”Positional Encoding” from a Transformer. It employs a collection ofsines and cosines at various frequencies (across the sequence). By definition, adjacent items will have position encodings that are comparable. The position encoding function is a stack of sines and cosines that vibrate at different frequencies depending on

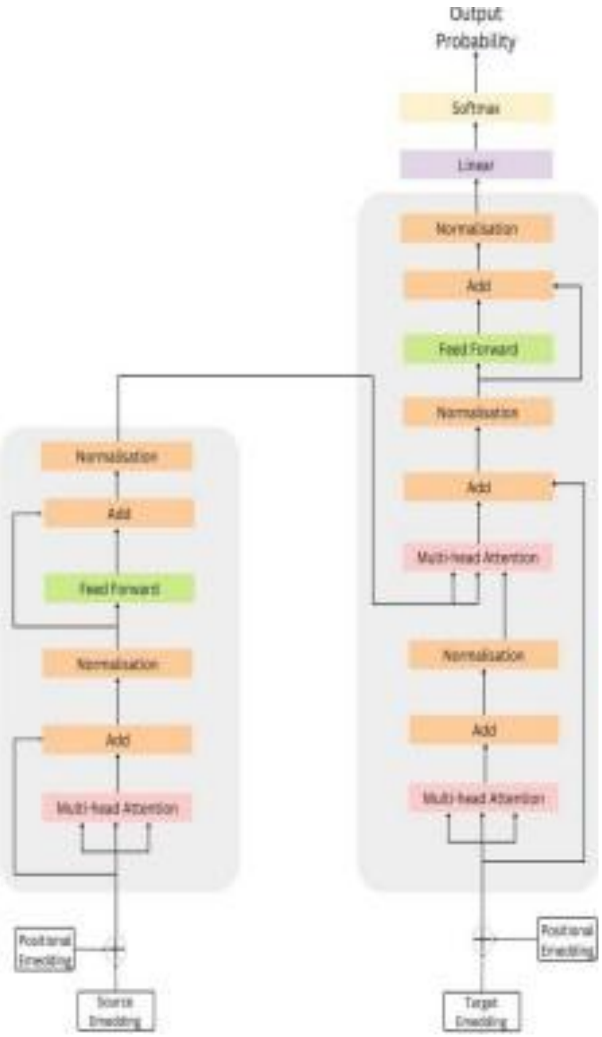
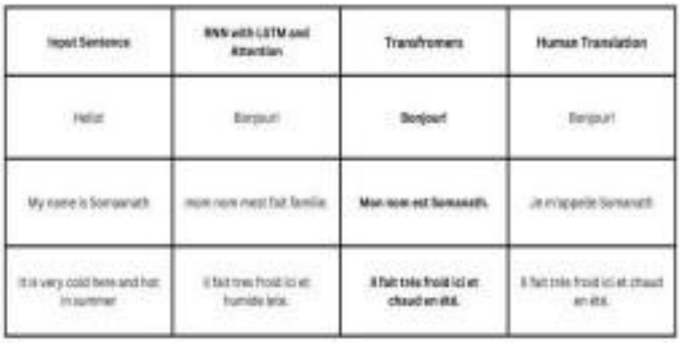


Fig. 6. The full structure of transformers.

B. Fine-tuning of Transformer Model

a) Transfer learning: Reusing a model that has already been trained on a different problem is known as transfer learning in machine learning. With transfer learning, a computer can use its understanding of one activity to better generalize about another [5]. During the training of this model, the middle hidden layers remain untrained and only the edge layers are trained. The first and last layers are trained during the training phase. New pre-training objectives, model architectures, unlabeled data sets, and other technologies have all contributed to recent advancements in transfer learning for NLP. To clarify the contributions and relevance of various strategies, we conduct an empirical analysis of them in this section. Since the field of transfer learning for NLP is one that is continually expanding, it is not practical for us to include every conceivable method or hypothesis in this empirical investigation



Input Sentence	RNN with LSTM and Attention	Transfromers	Human Translation
Hello!	Bonjour!	Bonjour!	Bonjour!
My name is Somanath	mon nom mest fait familie	Mon nom est Somanath.	Je m'appelle Somanath
It is very cold here and hot in summer	Il fait tres froid ici et humide leix.	Il fait très froid ici et chaud en été.	Il fait très froid ici et chaud en été.

Fig. 7. Comparision of results of NMT models(English to French translation). Clearly, we can see that transformer shows the best performance

b) Training Phase: The structure of transformers is shown in Fig.6. The transformer contains two main layers which are the encoder and decoder layer. Each layer contains different combination components. The main component of transformers is multi-head attention which takes input in the form which contains a query, key, and values. feed-forward network which is a dense layer. The pre-trained model is trained with English to French dataset

We employ an inverse square root learning rate schedule known as1

√

their location along the depth of the embedding vector. They vibrate across the position axis. The function that follows turns batches of text into a format appropriate for training.

- They are tokenized into ragged batches.
- Tokens are cut down so that it doesn't exceed max tokens.
- The target french tokens are divided into inputs and labels by this.
- These are shifted one step forward so that the label at each input point is the id of the subsequent token. • It gives back a pair of inputs and labels.
- It ensures that data is available when requested, prefetch lastly runs the dataset in parallel with the model.

the current training iteration and k denotes the number of warm-up steps (104 in all of our trials). For the first 104 steps, the learning rate is set to be constant at 0.01; after that, it ex ponentially declines until pre-training is complete. A triangle learning rate, which we also tried, generated marginally better results but necessitates knowing the whole number of training steps in advance. We select the more general inverse square root schedule since the number of training steps in some of our experiments will vary.

C. Evaluation of Model

As we discussed before the RNN didn't perform well for long sentences and its accuracy was also very low. These issues are hindered by transformer models. Transformer models produce good accuracy and a good Bleu score.Fig.7. show the performance comparison of the transformer. In all three

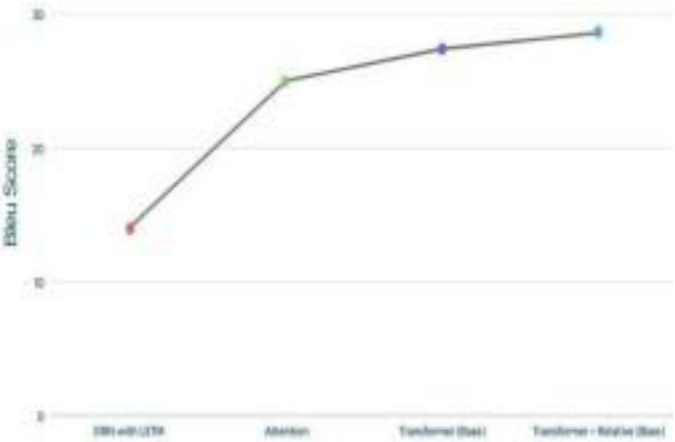


Fig. 8. Comparision of bleu score of NMT models(English to French translation). Clearly, we can see that transformer shows the best performance among the others

model configurations, Transformer clearly outperforms the competition. In particular, the gain is statistically significant for the zero-shot and multilingual models.

D. Model Deployment using Flask

~~max(n, k)~~during pre-training, where n denotes

serves as a web framework that makes it simple to create web apps. Its core is compact and simple to extend; it's a microframework without an object relationship manager or similar capabilities.

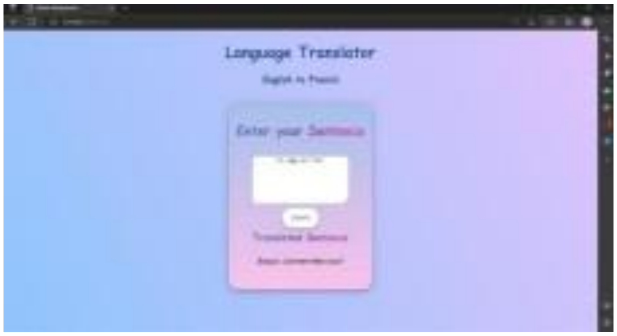


Fig. 9. User Interface created using HTML and CSS

b) Model Deployment: We created a user interface using HTML and CSS. Flask is used as a backend framework that connects the user interface and model pipeline. Flask frame work loads the user interface. The input text is transferred to the model pipeline. It will translate English to French and again it will be displayed in the user interface. Fig.8. Shows

a) Flask Framework: A Python package called Flask the workflow of our application.

Vi. Conclusion

In this project, we initially developed our project using Recurent Neural Network with an attention mechanism and it shows decent bleu score to small text but for long sentences the performance of this model is poor.So we moved to transformer to develop a language translator.Transformer model used a cutting-edge structure in its design, which significantly improved translation accuracy and speed.It shows good Bleu score for both long and short sentences.Without a doubt, Transformer is a vast advance over seq2seq models based on RNN.However, it has several drawbacks of its own,Only text strings of a certain length can be handled by attention. Before being entered as input into the system, the text must be divided into a certain number of segments or chunks. Context is broken up by this text chuncking. For instance, if a sentence is broken down the middle, a lot of context is lost. In other words, the phrase or any other semantic border is ignored while the text is divided. Still

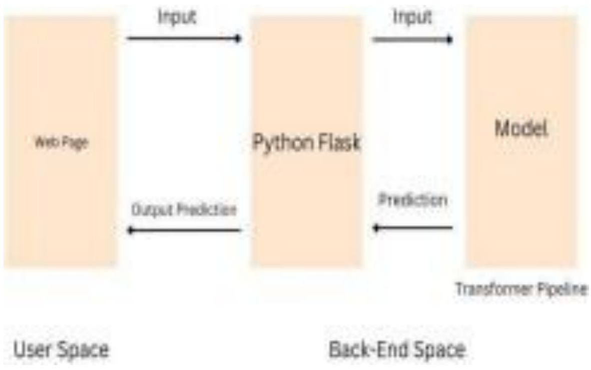


Fig. 10. The workflow of our application

Further reseach work required to enhancement of performance of transformer models.

References

[1] Luong MT, Pham H, Manning CD. Effective approaches to attention based neural machine translation. arXiv preprint arXiv:1508.04025. 2015 Aug 17.

[2] Yang S, Wang Y, Chu X. A survey of deep learning techniques for neural machine translation. arXiv preprint arXiv:2002.07526. 2020 Feb 18. [3] Bahdanau,D.,Cho, K.,Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.arXiv preprint arXiv:1409.0473. [4] Britz, D., Goldie, A.,Luong, M.T., Le, Q. (2017). Massive ex ploration of neural machine translation architectures.arXiv preprint arXiv:1703.03906.

[5] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res.. 2020 Jun;21(140):1-67.

[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30.