iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37687

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Neural Machine Translation Using Sequence Modeling

N. Revathi[1], Dr. R. Ravinder Reddy[2]
[1, 2]*Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Osmania University*

Abstract: Language is a main mode of communication, and translation is a critical tool for understanding information in a foreign language. Without the help of human translators, machine translation allows users to absorb unfamiliar linguistic material. The main goal of this project is to create a practical language translation from English to Hindi. Given its relevance and potential in the English-Hindi translation, machine translation is an efficient way to turn content into a new language without employing people. Among all available translation machines, Neural Machine Translation (NMT) is one of the most efficient ways. So, in this case, we're employing Sequence to Sequence Modeling, which includes the Recurrent Neural Network (RNN), Long and Short Term Memory (LSTM), and Encoder-Decoder methods. Deep Neural Network (DNN) comprehension and principles of deep learning, i.e. machine translation, are disclosed in the field of Natural Language Processing (NLP). In machine reclining techniques, DNN plays a crucial role.

Keywords: Sequence to Sequence, Encoder-Decoder, Recurrent Neural Network, Long & Short term Memory, Deep Neural Network.

I. INTRODUCTION

Machine Translation (MT) is a fully automated software that can convert source text into target languages. Humans can use machine translation (MT) to assist them in translating text and speech into another language, or the MT software can work independently. Statistical Machine Translation (SMT), Rule-based Machine Translation (RBMT), Hybrid Machine Translation (HMT), and Neural Machine Translation (NMT) are the four forms of machine translation. Machine translation has been in research since 1940. Machine translation is a language learning aid for non-native speakers. Many populated countries, like China and India, have multiple languages that shift region after region. India has, For example, 23 official languages recognised constitutionally and countless unofficial local languages (e.g. Hindi, Telugu). In large countries, linguistic variety is not simply abundant; but also small countries. In Papua New Guinea, one of the smallest populous areas, there are 851 languages spoken. There is an estimated three billion people in India, yet only approximately 10 percent can speak English.

In certain surveys, just 2% of persons who speak English are capable of talking, writing and studying English properly, while 8% can only recognise simple English and speak with different accents. A great number of useful materials are available on the Internet in English and the majority of the people in India cannot grasp this well. To enhance people's understanding, it is necessary that such information is translated into neighbourhood languages. It is crucial not just for business goals, but even to shared emotions, reviews and actions that information is shared between people. In order to minimise the communication gap between various people, translation plays an important function. Translation, It is not feasible to translate them manually considering the enormous amount of text. It is therefore vital that text is automatically translated from one language to another (say, English) (say, Tamil, Malayalam). Also known as machine translation is this technology. The challenges of morphological and structural differences are English to Indian translation. For example, (i) parallel corpora number; and (ii) language difference mostly due to syntactic divergence in terms of morphology and word order variance.

II. LITERATURE SURVEY

Machine translation is a discipline of computational linguistics with the goal of using a computer to mechanically translate text from one language to another. To the best of our knowledge, Petr Petrovich Troyanskii was the first person to formally present machine translation. In this review, we look at two important aspects of machine translation: statistical machine translation and neural machine translation. In most statistical approaches to machine translation, the phrase transition model is the most crucial component. The methods described above use neural networks to model continuous representations of language units. In domains like computer vision and speech recognition, deep neural networks have made substantial progress.

Statistics machine translation is one of the standard methods to solve the machine translation difficulty. This procedure requires large data sets for similar organised language pairs in grammar and is successful. In recent years, NMT has developed as an alternative method to the same problem. We are investigating a number of settings for the development of a Hindi-speaking machine translation system.

They examined and contrasted our findings with typical machine translation techniques with 8 separate combinations of NMT architectures, from English to Hindi. In their studies, they also found that NMT required less training data and consequently only a few thousand sentences with an adequate translation. Most machine translation systems in the preceding two decades were based on a statistical approach to machine translation. Sentences and phrases are the essential units of translation in these systems. These sentences can be one or more words long. Bayesian inferencing estimates the predicted probability of translation in pairs of words for most standard translation systems. One sentence is in the source language, while the other is in the target language in these pairs. With the previous procedures, combining and predicting the appropriate pair is rather difficult due to the extremely low likelihood of these sentences. The possibility of a certain couple of phrases increased one of the feasible methods. In addition, Google has updated its research on neural machine translation in recent years (NMT). In order to achieve a succession of learning processes, Sutskever developed a long and short speech process (LSTM). This neural machine translation system is a network-based machine with 8 layers and 8 layers. An encoder in the NMT system uses a bidirectional recurred neural network (RNN) known as a decoder to encode the source phrase in a second RNN. To improve the system's efficiency, this decoder architecture might be developed with multiple layers. In most cases, neural machine translation necessitates a lot of computing power, therefore it's only a good option if we have enough time or computing power. Another problem with the previous NMT was inconsistency in the processing of rare words. Learning and inference were ineffective due to the sluggish availability of these inputs on the network. The LSTM models, as well as the eight layers of decoder and encoder, greatly reduce the likelihood of such errors. The system's third big flaw was that it forgot the words over a long period of time. An 8-layer approach is used to handle this problem as well. After 2014, this study inspired many university students, and NMT is proving to be a viable alternative to machine translation's mainstream technology. The addition of an attention mechanism to neural machine translation models allows for selective inspection of source words, which improves machine translation system efficiency. This research has examined the efficiency of translation by use of bidirectional decoder attention models for morphologically rich language translation. For this analysis, the pair of English - Tamil has been used. Firstly, with the application of the incorporation in English as well as Tamil, translated results will increase by 0,73 BLEU points in the RNN Search baseline, with 4,84 BLEUs. The use of morphology before word vectorization, which divided the morphologically rich Tamil word into discrete morphemes before translation, resulted in an 8-fold reduction in target vocabulary. Furthermore, the performance of the RNN-Morph neural translation machine improved by 7.05 BLEU points over RNNSearch used on a single body. Since RNN-Morph model BLEU evaluations may be incorrect because of the growth in the number of matching tokens each sentence, human assessment measurements of sufficiency, fluidity and relative rankings also compared translation performance. In addition, the application of morphological segmentation boosted the mechanism's effectiveness.

III. PROPOSED SYSTEM

We used NMT on two of India's most morphologically rich languages, English and Hindi. For low-resource, morphologically rich Indian languages with limited online translation resources, we proposed a novel NMT model combining Multihead self-attention with pre-trained Byte-Pair-Encoded (BPE) and Multi-BP-Embeddings to develop an efficient translation system that overcomes the OOV (Out Of Vocabulary) Problem. We also gathered data from a variety of sources, resolved flaws with publicly available data, and polished it for future use. The BLEU score was utilised to assess the performance of our system.

IV. FLOW CHART OF MACHINE TRANSLATION

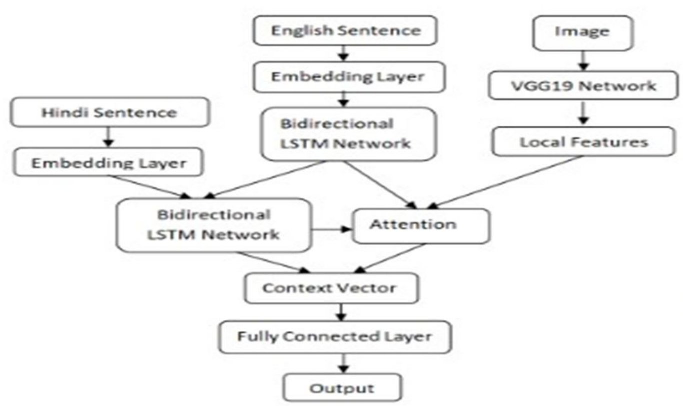


Fig 4.1 Flowchart Of Machine Translation

Because of its success, Neural Machine Translation (NMT) is a revolutionary, highly active approach to machine translation that has demonstrated promising results and attracted a huge number of academics to the subject. We look at how the architecture of NMTs evolved over relatively short period of time in this study. Starting with the basic encoder-decoder architecture, which had two flaws: poor performance with longer phrases and a difficulty with out-of-vocabulary phrases (OOV). Attention-based NMT works better with longer sentences, although it still has the OOV problem. Attention-based NMT, as well as sub-word segmentation, known as Sub-word NMT, are employed to address the OOV problem. NMT systems can cover a wider vocabulary range with word segmentations.

V. METHODOLOGY

1) *Neural Machine Translation:* NMT stands for Neural Machine Translation, which is a machine translation system that improves the fluency and accuracy of the translation process by using an artificial neural network. A basic encoder-decoder network underpins the NMT system. The neural networks used in NMT are recurrent neural networks (RNN). The basic architecture of the RNN is the reason for choosing it for the assignment. RNNs feature a cyclic shape that facilitates learning repeated sequences easier than other networks. RNN can be unrolled to store phrases as a series in both the source and destination languages. This demonstrates how a single layer may be unrolled into multiple layers and how prior time period data can be kept in a single cell. When the alignment between the inputs and outputs is known ahead of time, RNNs can readily map sequences to sequences.
2) *Recurrent Neural Networks:* A Recurrent Neural Network (RNN) is a type of Artificial neural network which uses sequential data or Time series data. RNN are the state of the art algorithm for sequential data. Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist. It has Vanishing, Exploding Gradient-Problem. To overcome this problem we are using here Long and Short term Memory.

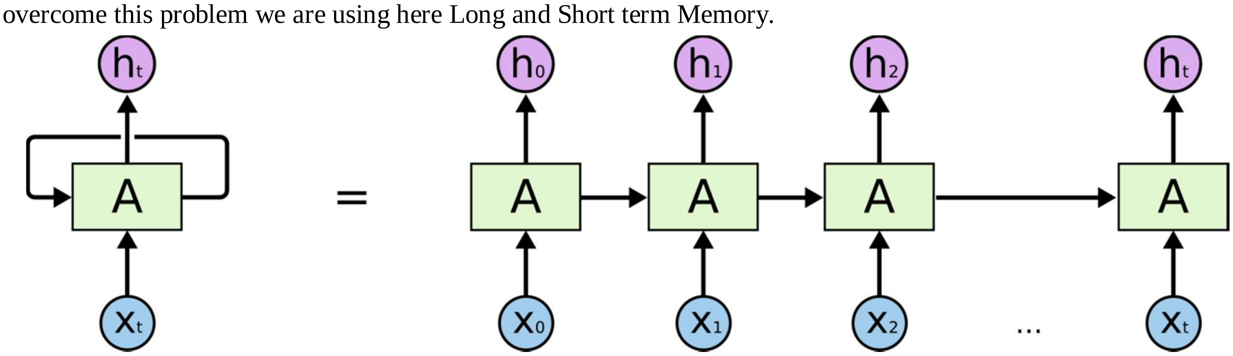


Fig 5.1 An unrolled Recurrent Neural Network

3) *LSTM Networks:* RNN consumes sentences word by word, updating the hidden state as each word is processed. Simple RNNs employ the tanh or sigmoid activation functions, which fail to grasp numerous long-term dependencies in sentences because they place greater emphasis on the most recent words encountered. During training, this also has the issue of vanishing and exploding gradients. LSTM learns to selectively forget and recall rather than storing every detail of a sentence in state. The gating mechanism is used to accomplish this.

It consists of three gates and a candidate memory cell. Gates aids in the selective forgetting and remembering of information in a sentence, with the contents being stored in a memory cell. The figure depicts the structure of an LSTM network. The layers of the LSTM are made up of the following components.

a) *Forget Gate (ft):* It chooses which information from the previous memory cell to delete.
b) *Input Gate (it):* It chooses which data to send to each cell to be remembered.
c) *Candidate Memory (C´ t):* It keeps track of the data from the current input.
d) *Output Gate (ot):* The current hidden state is computed using the tanh function, which squashes cell memory to lie between -1 and 1. At this moment, the output gate determines what should be output in the hidden state.
e) *Cell Memory (Ct):* It is the real cell memory that has been updated after relevant information has been added and unneeded data has been removed.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

LSTMs are specifically developed to prevent the problem of long-term dependency.

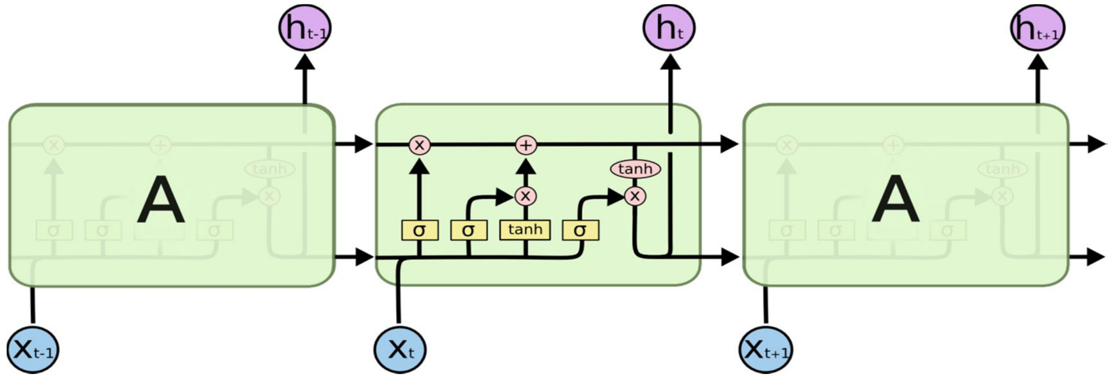


Fig 5.2 The LSTM contains four interacting layers

LSTMs have a chain-like structure as well, but the repeating module is different. Instead of one neural network layer, there are four, each interacting in a unique way.

4) *Bi-directional Long and Short Term Memory:* Bidirectional LSTMs are a type of LSTM that can be used to increase model performance in sequence classification issues. These bidirectional LSTM will manage your inputs in two directions: from the past to the future and from the future to the past.

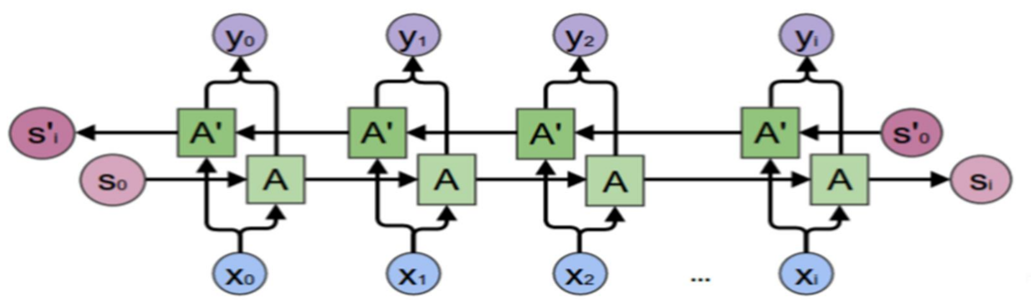


Fig 5.3 Bidirectional LSTM

5) *Encoder and Decoder:* The Encoder-Decoder model is a way of using recurrent neural networks for Sequence-to-Sequence prediction problems. The approach involves two Recurrent neural networks, one to encode the input sequence, called the Encoder and a second to decode the encoded input sequence into the target sequence called the Decoder.

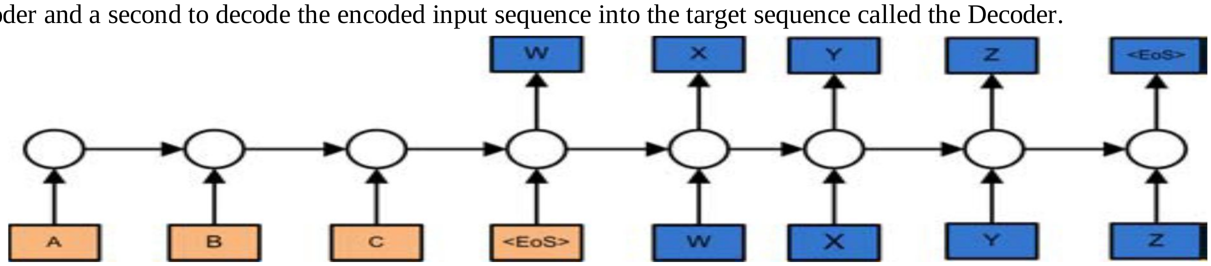


Fig 5.4 Sentence modeling in LSTM network

6) *Embedding Layer:* The Embedding layer is defined as the first hidden layer of a network. Word embeddings can be thought of as an alternate to one-hot encoding along with dimensionality reduction. It enables us to convert each word into a fixed length vector of defined size.
7) *Attention:* Attention mechanisms are being increasingly used to improve the performance of Neural Machine Translation (NMT) by selectively focusing on sub-parts of the sentence during translation. Attention in neural machine translation provides the possibility to encode relevant parts of the source sentence at each translation step. As a result, attention is considered to be an alignment model as well.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

VI. IMPLEMENTATION

A. We processed the dataset so as to remove the punctuations present there using regular expression library.
B. Convert the dataset into Comma Separated Values (CSV) and import the libraries.
C. After that we need to do Train the dataset Test and Split the dataset.
D. Then Here, we are using Sequence to Sequence Model. So, mainly Recurrent Neural Network (RNN) , Long and Short Term Memory (LSTM), and also Encoder-Decoder method are used.
E. Spell Corrections for English words: **-** Using autocorrect library in order to make corrections, if any Misspelled words are present.
F. Convert all upper-case letter to lower case counterparts in order to make our dataset homogeneous.
G. There are words (may be few or more) which are not present in the embedding file. This steps basically deals with finding those words.
H. The few words that were discovered to be missing in the preceding phase are largely nouns. As a result, we simply replaced them with the letter ‘a' (as flexible as our wish).
I. We used Integer encoding for Hindi words by using sklearn Label Encoder
J. We have assumed the maximum length of English and Hindi sentences to be of 30 words which will not be less than the length of any sentence present in our dataset.
K. Then we padded all those English and Hindi sentences which were falling short from maximum length of 30 words.
L. Implementing LSTM for English sentences length and Hindi sentences due to Vanishing Gradient and Exploding Gradient problem.
M. At the end, we computed BLEU Score for Testing and Training.

DATASET

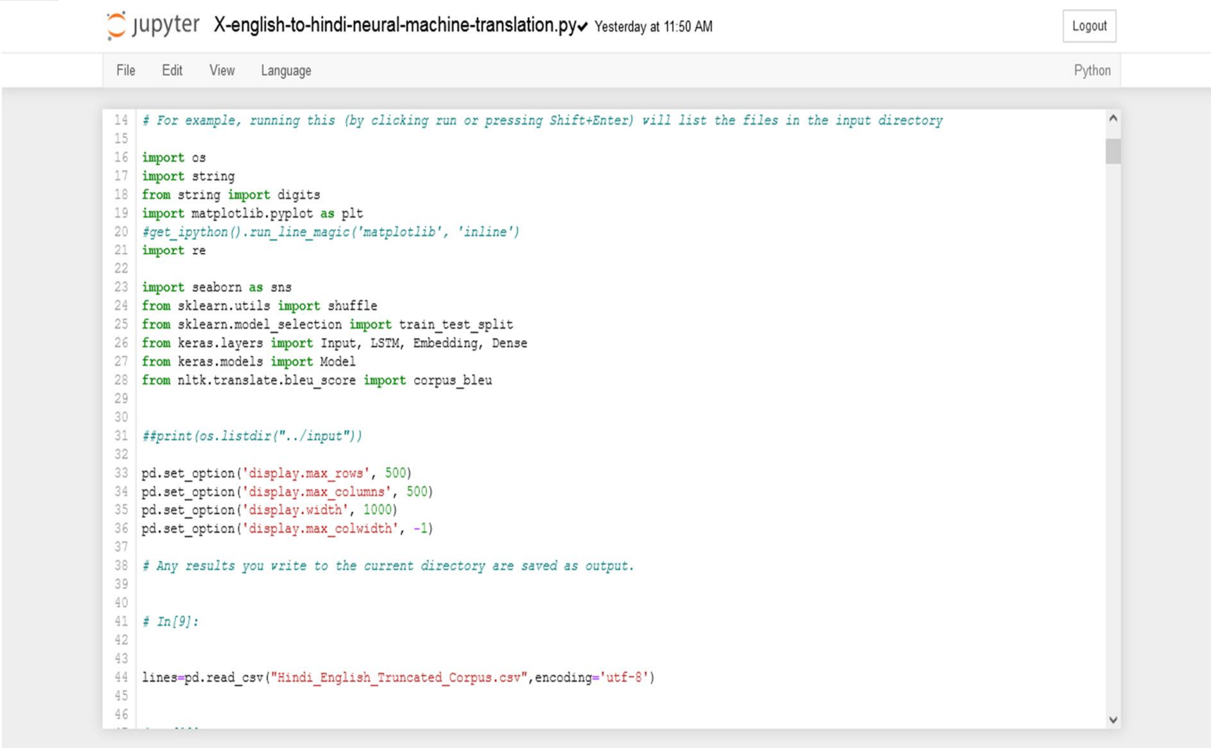
politicians do not have permission to do what needs to be done.	राजनीतिज्ञों के पास जो कार्य करना चाहिए, वह करने कि अनुमति नहीं है .
I'd like to tell you about one such child,	मई आपको ऐसे ही एक बच्चे के बारे में बताना चाहूंगी,
This percentage is even greater than the percentage in India.	यह प्रतिशत भारत में हिन्दुओं प्रतिशत से अधिक है।

Fig 6.1 Sample records from the dataset

VII. RESULTS AND DISCUSSIONS

In Machine Translation mostly the quality of our work is measured by the BLEU Score. BLEU (Bi-Lingual Evaluation Understudy) is a metric for assessing machine-translated text automatically. The BLEU score is a value between 0 and 1 that indicates how closely the machine-translated text resembles a set of high-quality reference translations. The Bilingual Evaluation Understudy Score (BLEU) is a statistic used to compare a generated sentence to a reference sentence. The score was established to evaluate the accuracy of predictions made by automatic machine translation systems. For the example statement, “Once you stop learning, you start dying,” an n-gram is a sequence of words that occur within a particular window, where n specifies the window size...unigram, bigram, and trigram. To count the number of matches, BLEU compares the n-gram of the candidate translation to the n-gram of the reference translation. The BLEU metric assigns a score to a translation on a scale of 0 to 1, attempting to assess the MT output's adequacy and fluency. The closer a test sentence's score is to one, the more similar their human reference translations are. The BLEU score is a measure of a model's overall quality. BLEU could stand for "blue" in French.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

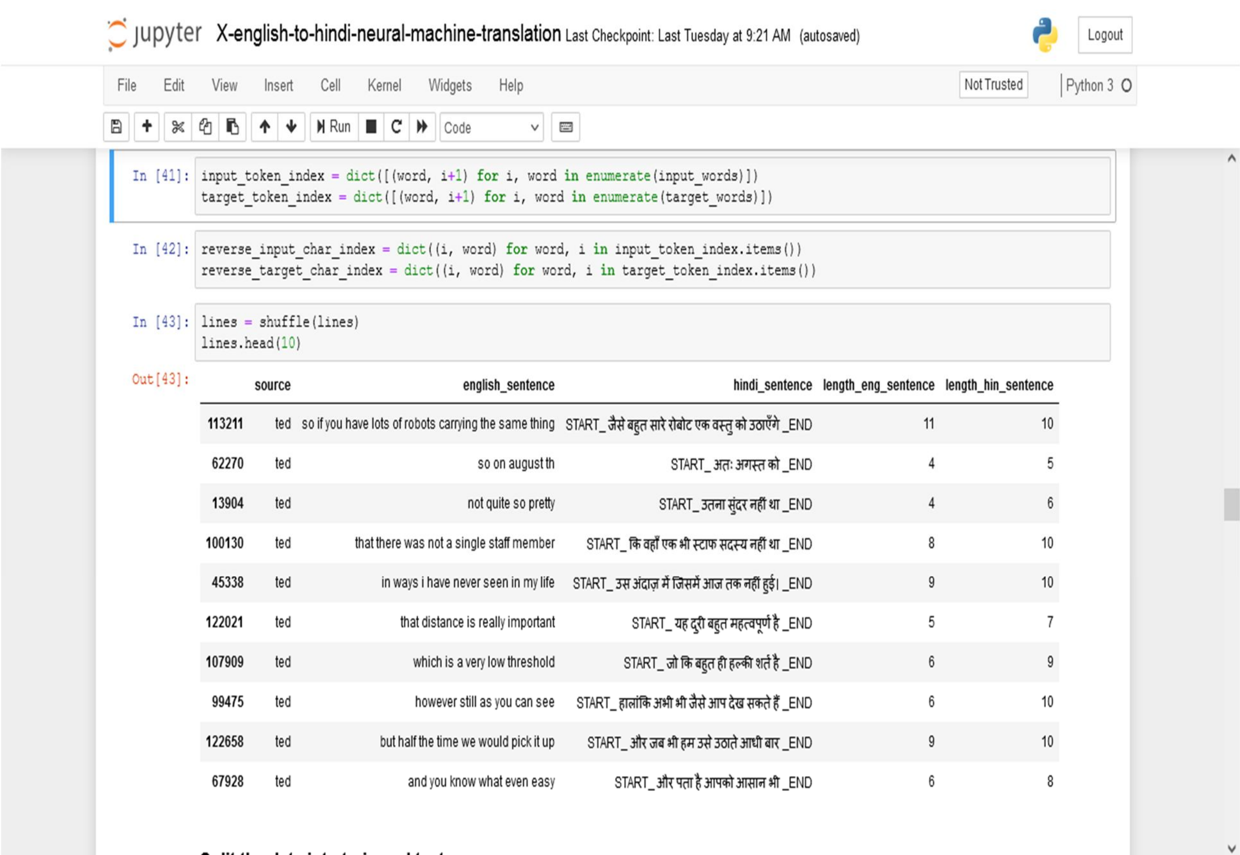


jupyter X-english-to-hindi-neural-machine-translation.py✔ Yesterday at 11:50 AM Logout

File Edit View Language Python

```python
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os
import string
from string import digits
import matplotlib.pyplot as plt
#get_ipython().run_line_magic('matplotlib', 'inline')
import re

import seaborn as sns
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from keras.layers import Input, LSTM, Embedding, Dense
from keras.models import Model
from nltk.translate.bleu_score import corpus_bleu


##print(os.listdir("../input"))

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
pd.set_option('display.max_colwidth', -1)

# Any results you write to the current directory are saved as output.


# In[9]:


lines=pd.read_csv("Hindi_English_Truncated_Corpus.csv",encoding='utf-8')
```

Fig 7.1 Importing libraries



jupyter X-english-to-hindi-neural-machine-translation Last Checkpoint: Last Tuesday at 9:21 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In [41]:
```python
input_token_index = dict([(word, i+1) for i, word in enumerate(input_words)])
target_token_index = dict([(word, i+1) for i, word in enumerate(target_words)])
```

In [42]:
```python
reverse_input_char_index = dict((i, word) for word, i in input_token_index.items())
reverse_target_char_index = dict((i, word) for word, i in target_token_index.items())
```

In [43]:
```python
lines = shuffle(lines)
lines.head(10)
```

Out[43]:

	source	english_sentence	hindi_sentence	length_eng_sentence	length_hin_sentence
113211	ted	so if you have lots of robots carrying the same thing	START_ जैसे बहुत सारे रोबोट एक वस्तु को उठाएँगे _END	11	10
62270	ted	so on august th	START_ अत: अगस्त को _END	4	5
13904	ted	not quite so pretty	START_ उतना सुंदर नहीं था _END	4	6
100130	ted	that there was not a single staff member	START_ कि वहाँ एक भी स्टाफ सदस्य नहीं था _END	8	10
45338	ted	in ways i have never seen in my life	START_ उस अंदाज़ में जिसमें आज तक नहीं हुई। _END	9	10
122021	ted	that distance is really important	START_ यह दुरी बहुत महत्वपूर्ण है _END	5	7
107909	ted	which is a very low threshold	START_ जो कि बहुत ही हल्की शर्त है _END	6	9
99475	ted	however still as you can see	START_ हालांकि अभी भी जैसे आप देख सकते हैं _END	6	10
122658	ted	but half the time we would pick it up	START_ और जब भी हम उसे उठाते आधी बार _END	9	10
67928	ted	and you know what even easy	START_ और पता है आपको आसान भी _END	6	8

Fig 7.2 Read the Dataset

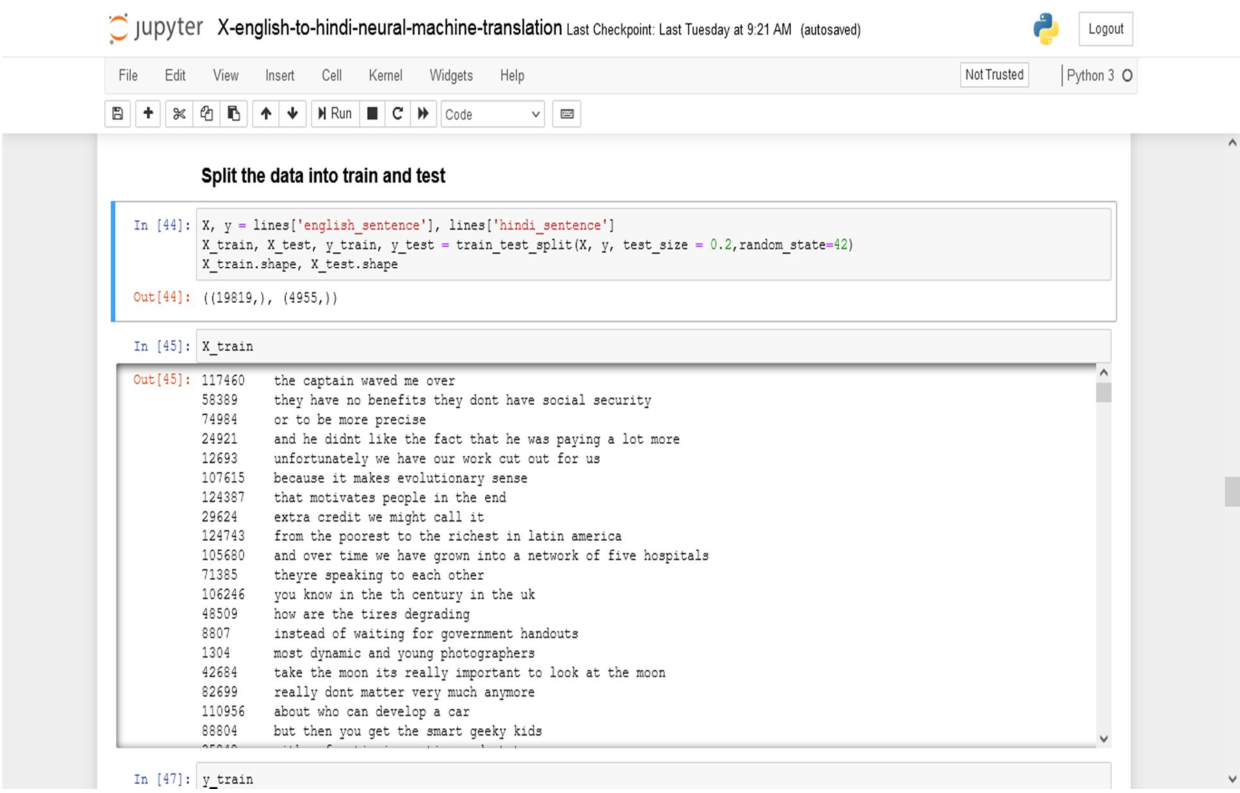


Fig 7.3 Splitting the data

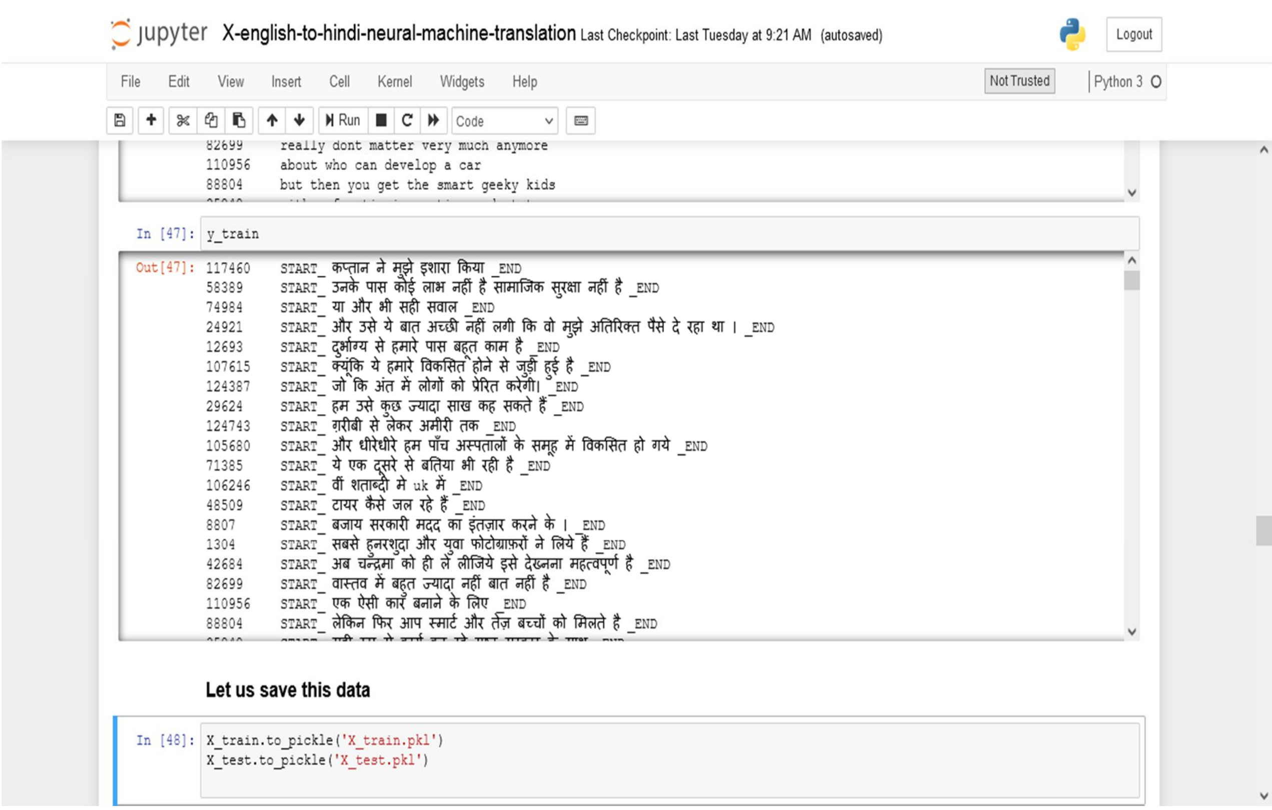


Fig 7.4 Train the data

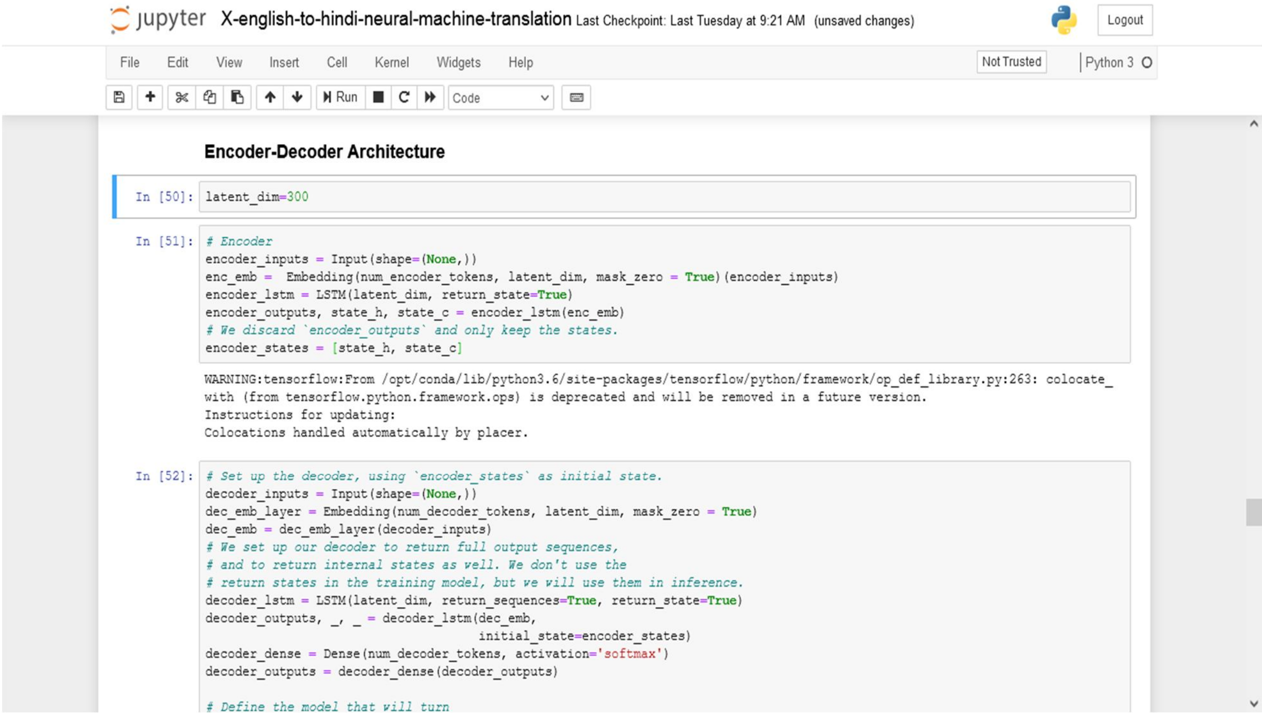


Fig 7.5 Encoder-Decoder

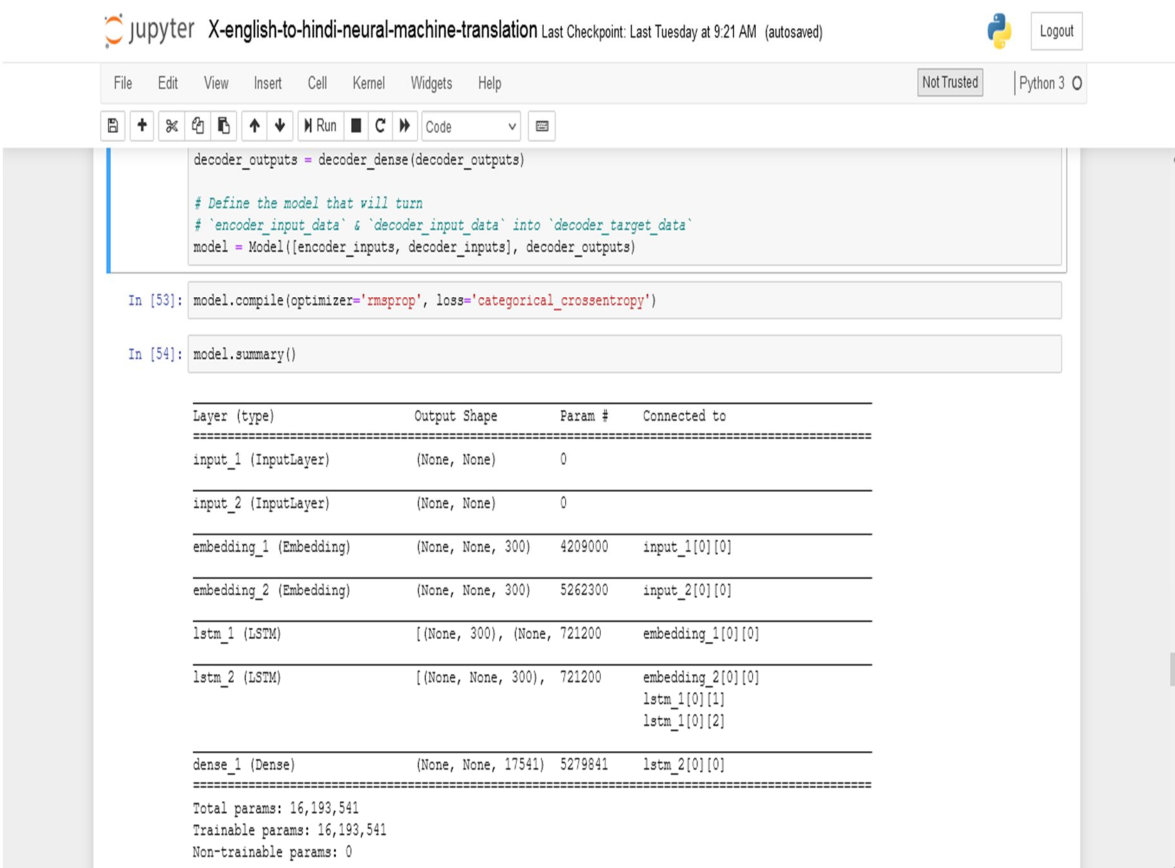


Fig 7.6 Trained model

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

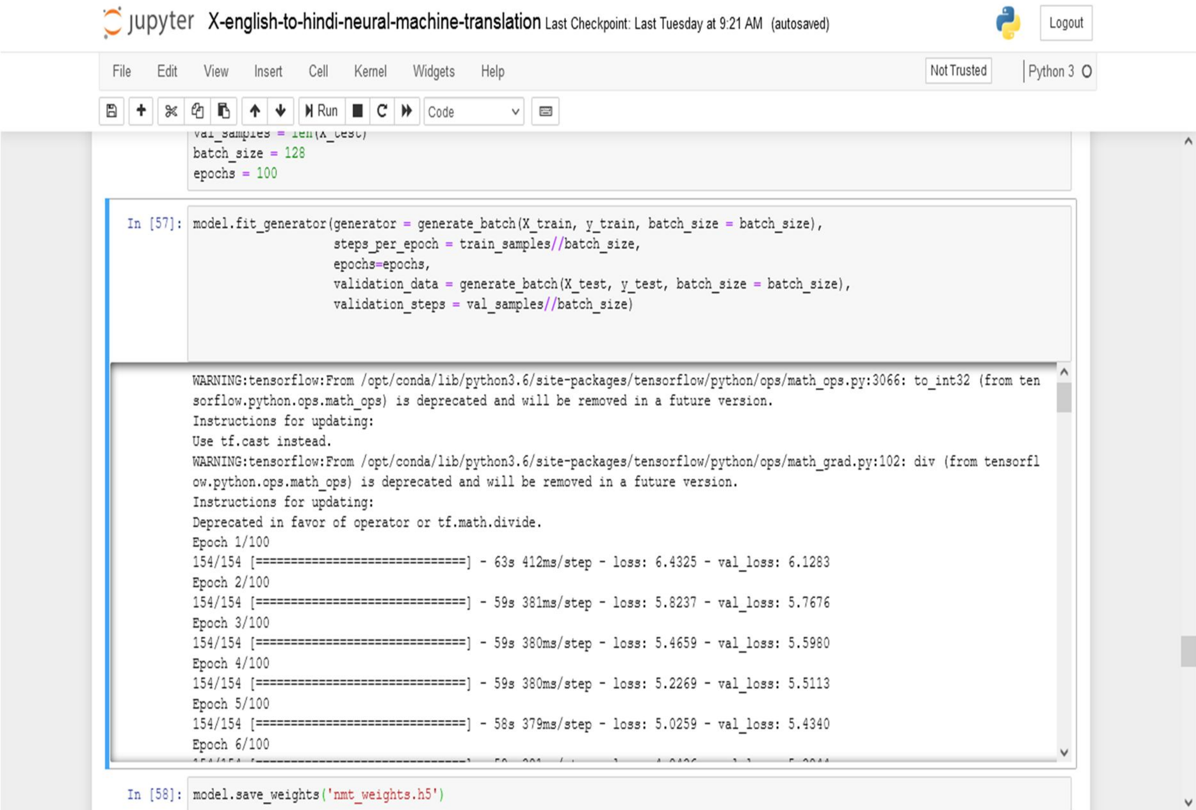


Fig 7.7 Model training with epochs

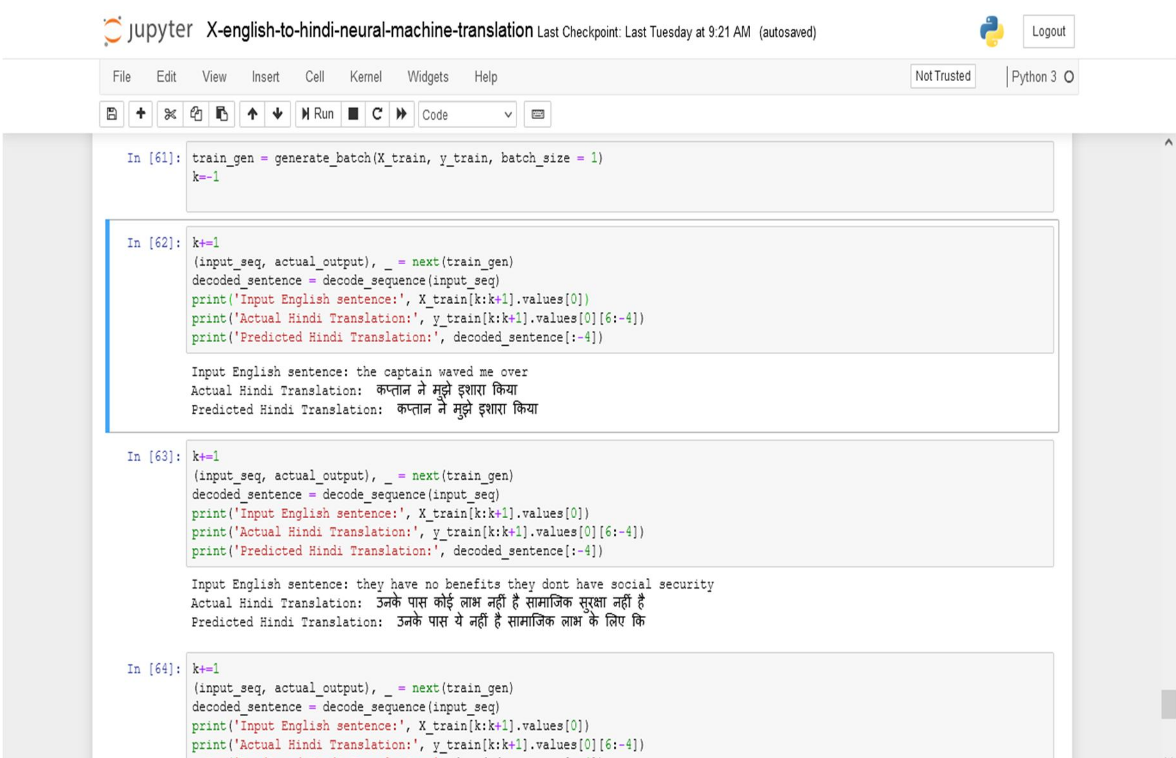
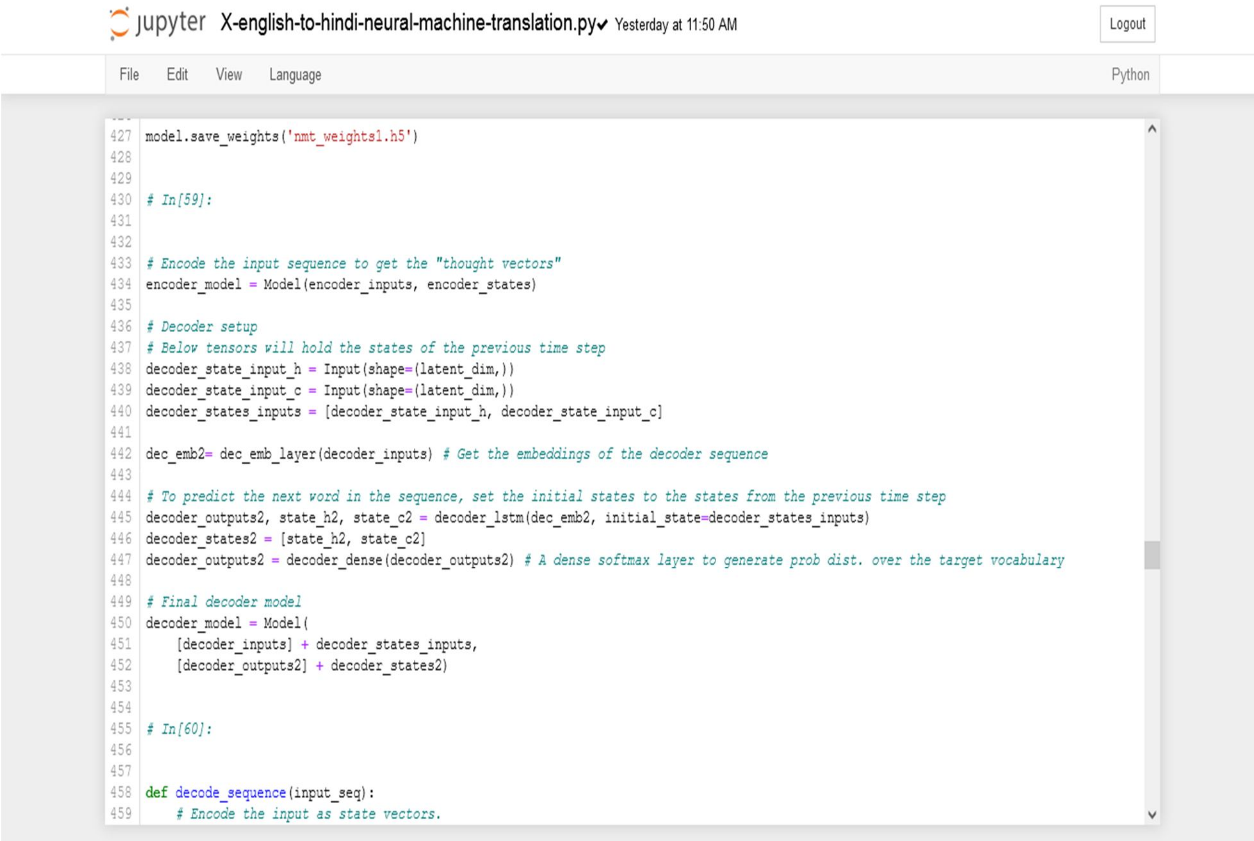


Fig 7.8 MT Output

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

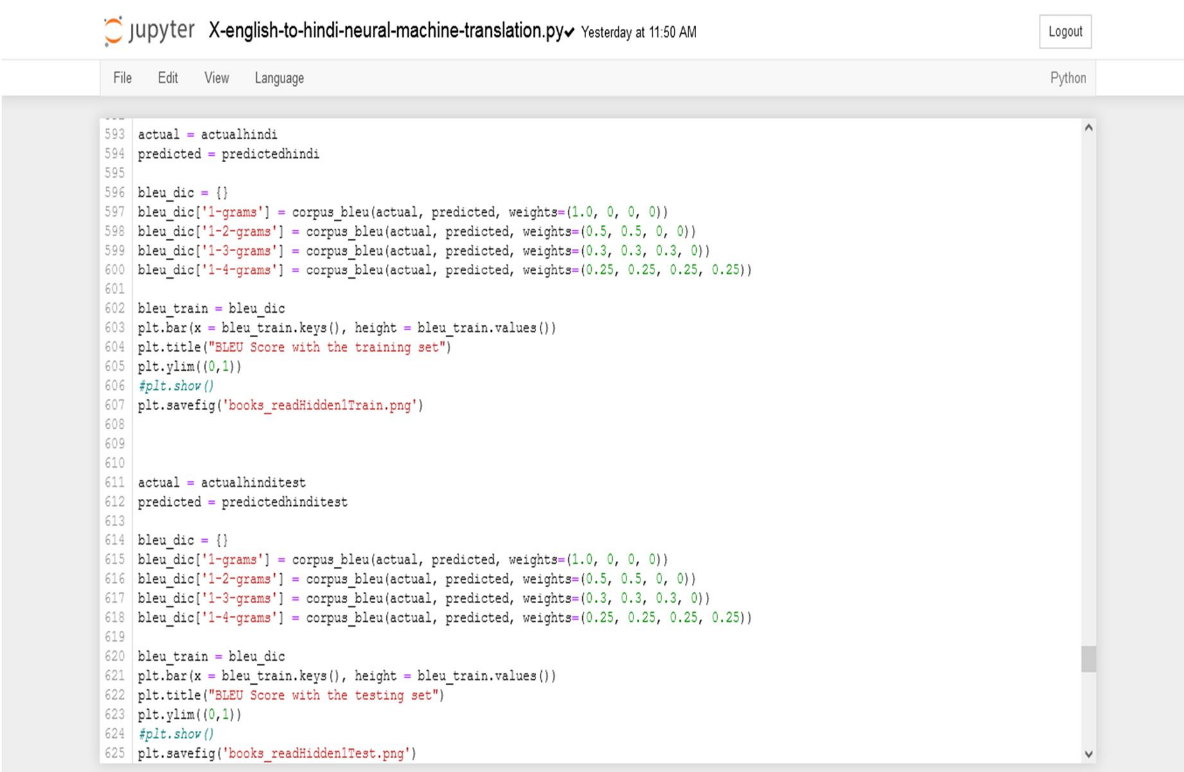


jupyter X-english-to-hindi-neural-machine-translation.py✔ Yesterday at 11:50 AM Logout

File Edit View Language Python

```python
model.save_weights('nmt_weights1.h5')


# In[59]:


# Encode the input sequence to get the "thought vectors"
encoder_model = Model(encoder_inputs, encoder_states)

# Decoder setup
# Below tensors will hold the states of the previous time step
decoder_state_input_h = Input(shape=(latent_dim,))
decoder_state_input_c = Input(shape=(latent_dim,))
decoder_states_inputs = [decoder_state_input_h, decoder_state_input_c]

dec_emb2= dec_emb_layer(decoder_inputs) # Get the embeddings of the decoder sequence

# To predict the next word in the sequence, set the initial states to the states from the previous time step
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2, initial_state=decoder_states_inputs)
decoder_states2 = [state_h2, state_c2]
decoder_outputs2 = decoder_dense(decoder_outputs2) # A dense softmax layer to generate prob dist. over the target vocabulary

# Final decoder model
decoder_model = Model(
    [decoder_inputs] + decoder_states_inputs,
    [decoder_outputs2] + decoder_states2)


# In[60]:


def decode_sequence(input_seq):
    # Encode the input as state vectors.
```

Fig 7.9 Dense layer 1



jupyter X-english-to-hindi-neural-machine-translation.py✔ Yesterday at 11:50 AM Logout

File Edit View Language Python

```python
actual = actualhindi
predicted = predictedhindi

bleu_dic = {}
bleu_dic['1-grams'] = corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0))
bleu_dic['1-2-grams'] = corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0))
bleu_dic['1-3-grams'] = corpus_bleu(actual, predicted, weights=(0.3, 0.3, 0.3, 0))
bleu_dic['1-4-grams'] = corpus_bleu(actual, predicted, weights=(0.25, 0.25, 0.25, 0.25))

bleu_train = bleu_dic
plt.bar(x = bleu_train.keys(), height = bleu_train.values())
plt.title("BLEU Score with the training set")
plt.ylim((0,1))
#plt.show()
plt.savefig('books_readHidden1Train.png')



actual = actualhinditest
predicted = predictedhinditest

bleu_dic = {}
bleu_dic['1-grams'] = corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0))
bleu_dic['1-2-grams'] = corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0))
bleu_dic['1-3-grams'] = corpus_bleu(actual, predicted, weights=(0.3, 0.3, 0.3, 0))
bleu_dic['1-4-grams'] = corpus_bleu(actual, predicted, weights=(0.25, 0.25, 0.25, 0.25))

bleu_train = bleu_dic
plt.bar(x = bleu_train.keys(), height = bleu_train.values())
plt.title("BLEU Score with the testing set")
plt.ylim((0,1))
#plt.show()
plt.savefig('books_readHidden1Test.png')
```

Fig 7.10 Adding BLEU Score

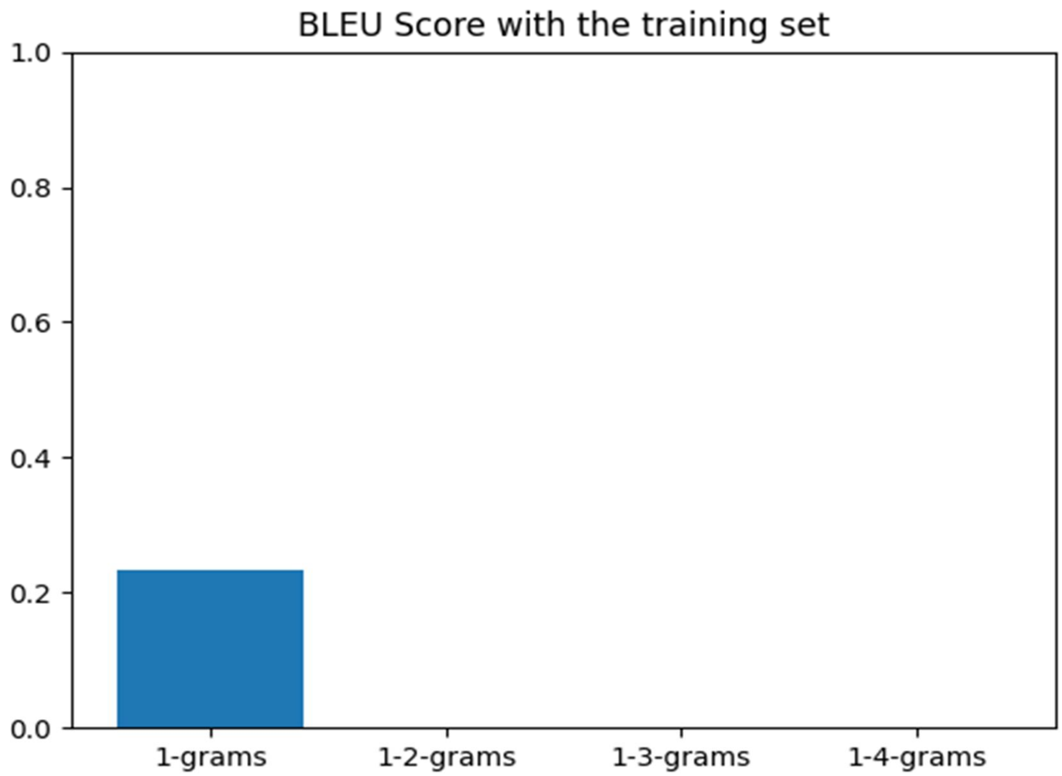


Fig 7.11 Dense layer 1 BLEU Score Training Set

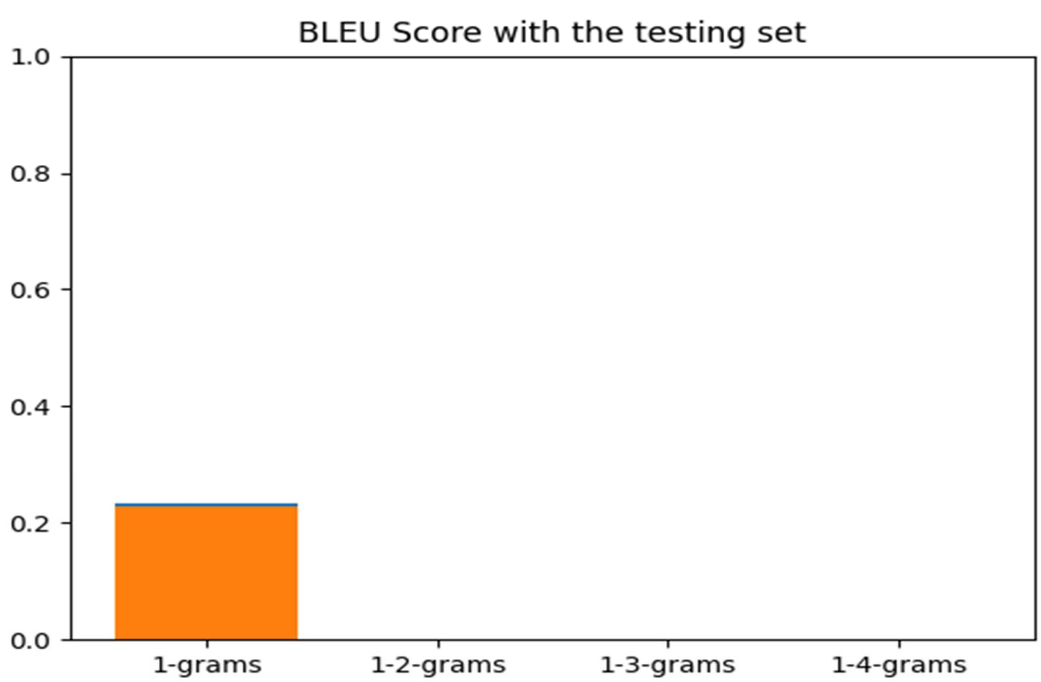


Fig 7.12 Dense layer 1 BLEU Score Testing Set

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

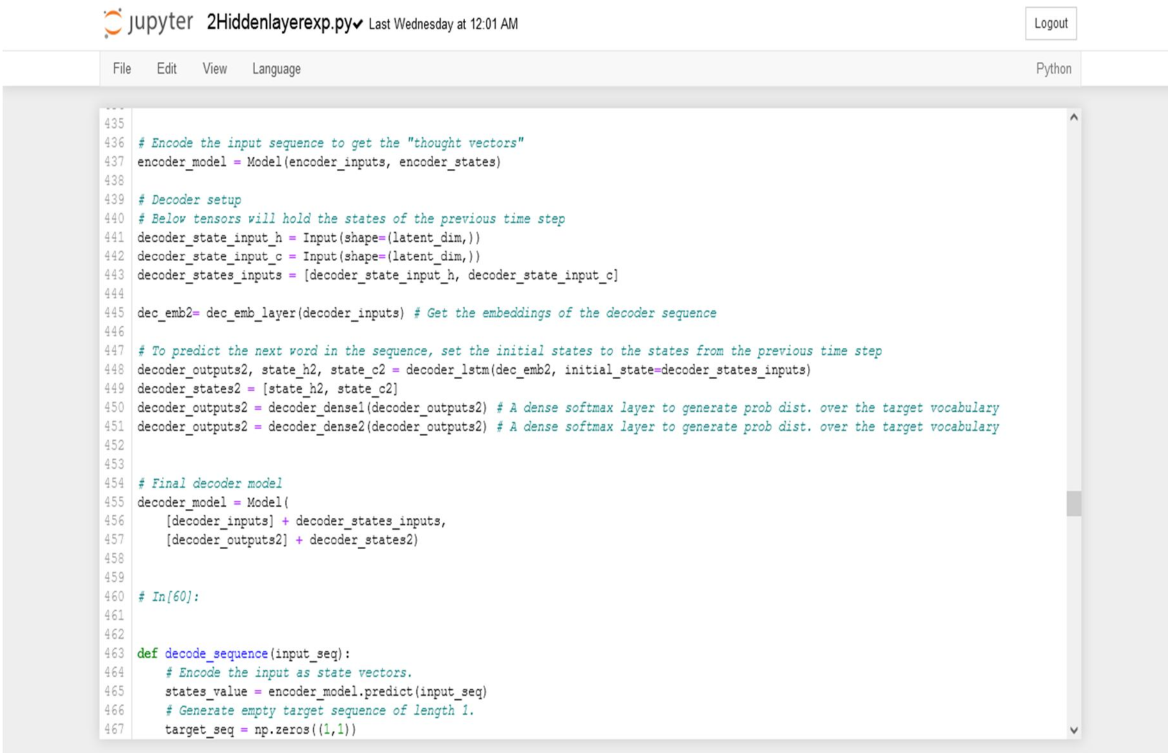


Fig 7.13 Dense Layer 2

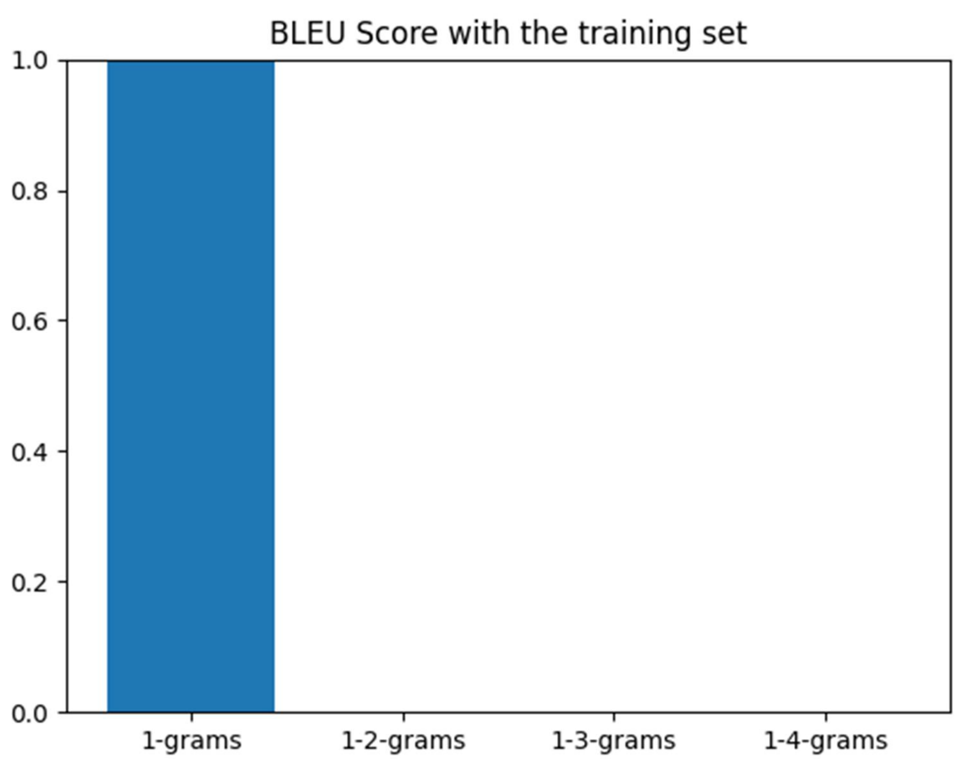


Fig 7.14 Dense layer 2 BLEU Score Training Set

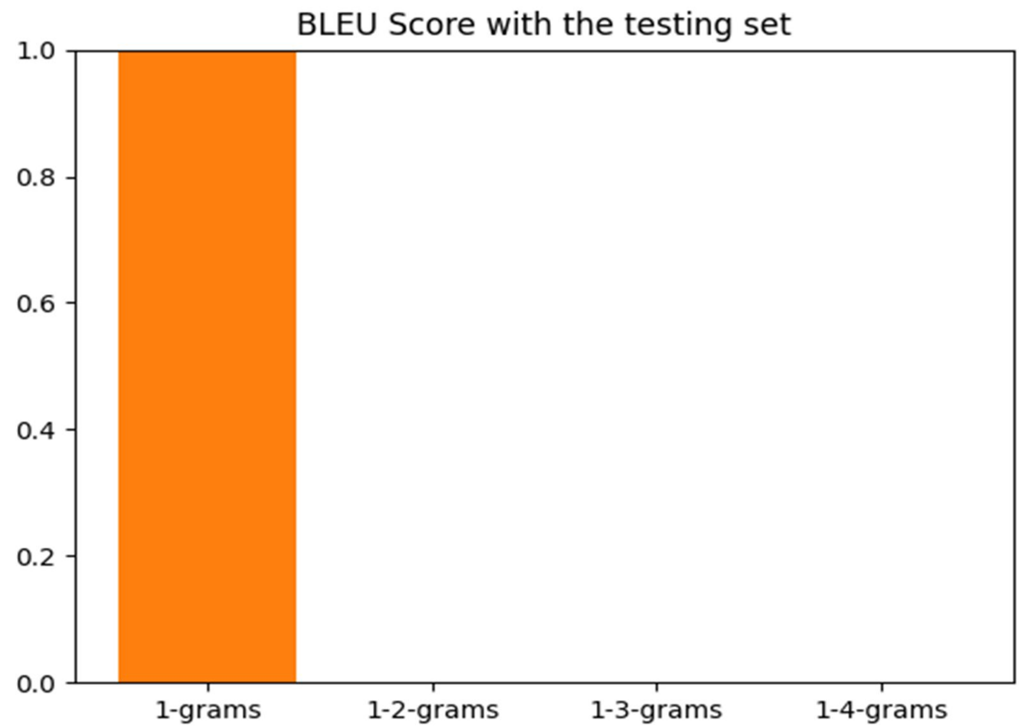
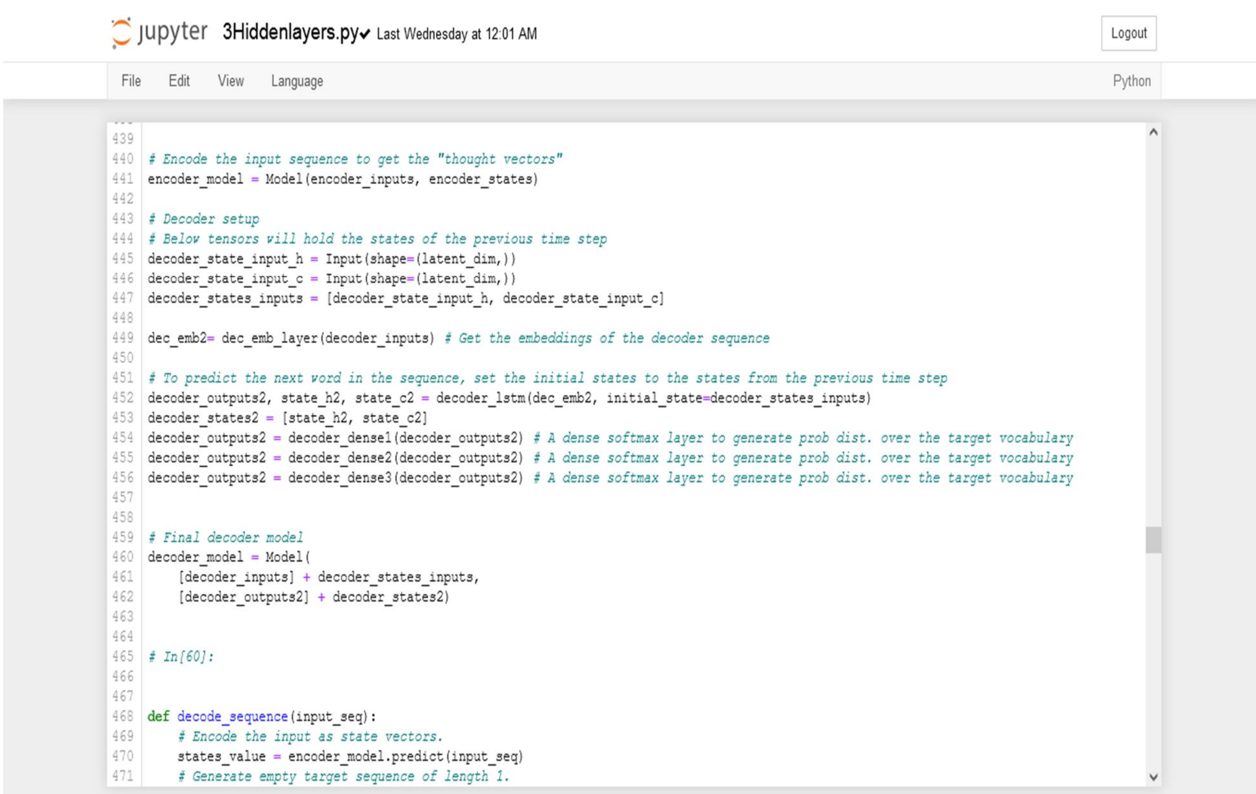


Fig 7.15 Dense layer 2 BLEU Score Testing Set




```
# Encode the input sequence to get the "thought vectors"
encoder_model = Model(encoder_inputs, encoder_states)

# Decoder setup
# Below tensors will hold the states of the previous time step
decoder_state_input_h = Input(shape=(latent_dim,))
decoder_state_input_c = Input(shape=(latent_dim,))
decoder_states_inputs = [decoder_state_input_h, decoder_state_input_c]

dec_emb2= dec_emb_layer(decoder_inputs) # Get the embeddings of the decoder sequence

# To predict the next word in the sequence, set the initial states to the states from the previous time step
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2, initial_state=decoder_states_inputs)
decoder_states2 = [state_h2, state_c2]
decoder_outputs2 = decoder_dense1(decoder_outputs2) # A dense softmax layer to generate prob dist. over the target vocabulary
decoder_outputs2 = decoder_dense2(decoder_outputs2) # A dense softmax layer to generate prob dist. over the target vocabulary
decoder_outputs2 = decoder_dense3(decoder_outputs2) # A dense softmax layer to generate prob dist. over the target vocabulary


# Final decoder model
decoder_model = Model(
    [decoder_inputs] + decoder_states_inputs,
    [decoder_outputs2] + decoder_states2)


# In[60]:


def decode_sequence(input_seq):
    # Encode the input as state vectors.
    states_value = encoder_model.predict(input_seq)
    # Generate empty target sequence of length 1.
```

Fig 7.16 Dense layer 3

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

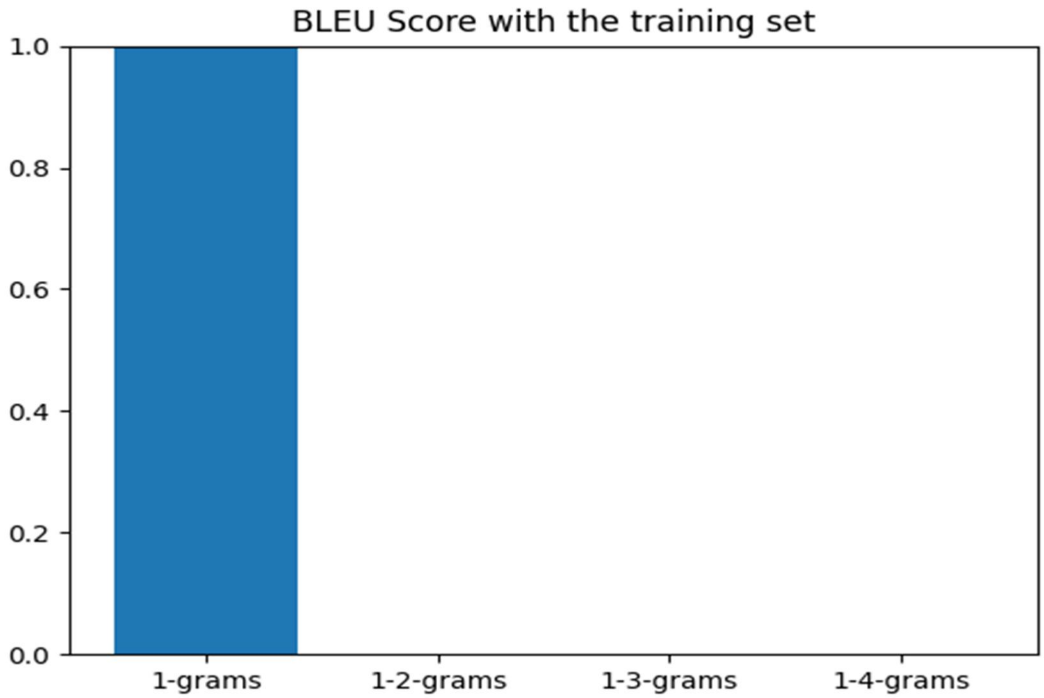


Fig 7.17 Dense layer 3 BLEU Score Training Set

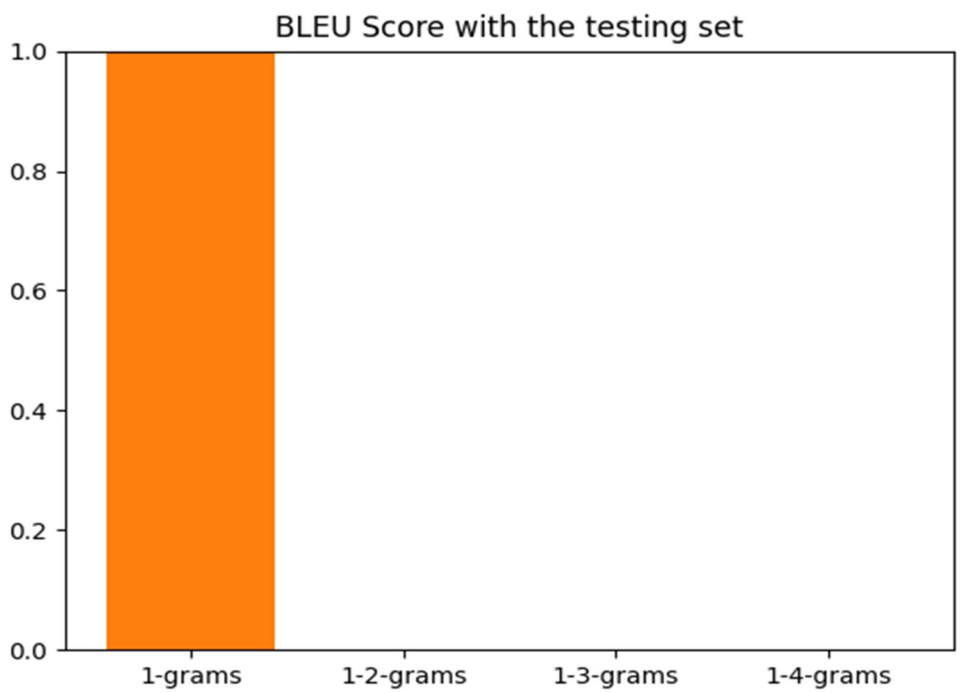


Fig 7.18 Dense layer 3 BLEU Score Testing Set

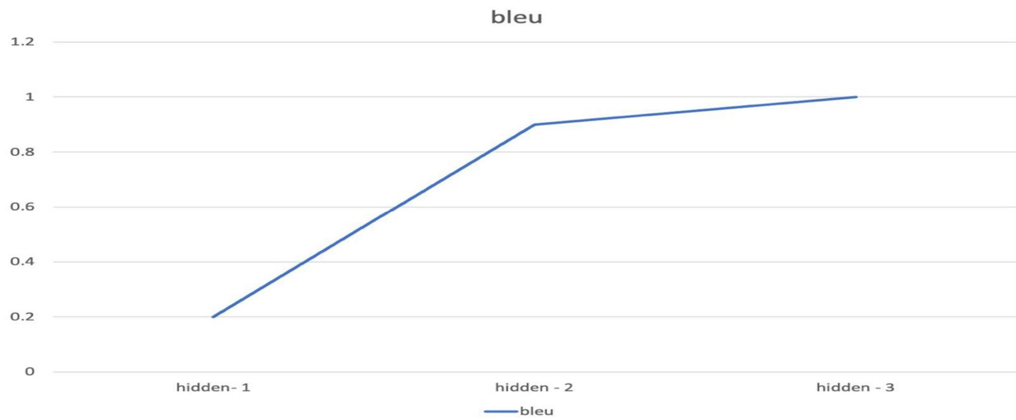


Fig 7.19 BLEU Graph

BLEU Score	Epochs	RNN	LSTM	Bi-LSTM
1	10	0.2	0.5	0.7
2	20	1.3	1.5	1.8
3	30	2.2	2.3	2.4
4	40	3.4	3.6	3.9
5	50	4.1	4.5	4.7

Fig 7.20 Tabular form of Results

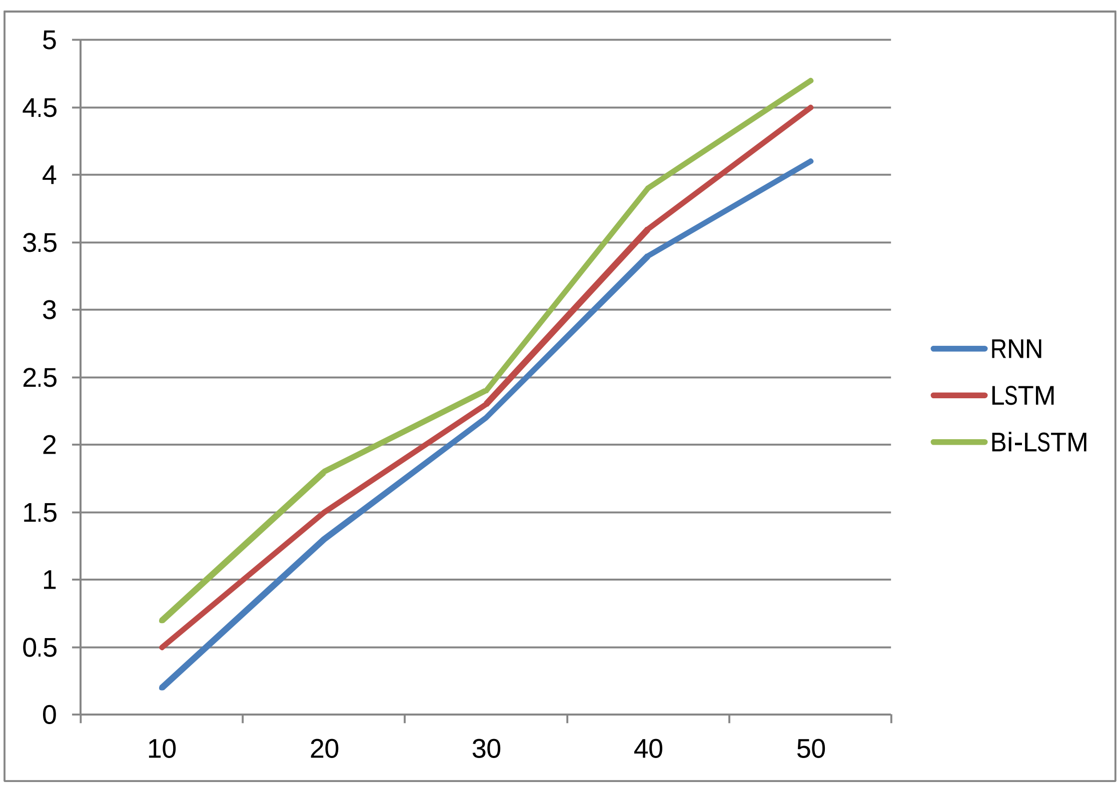


Fig 7.21 Graph overview

VIII. CONCLUSION

For a long time, Statistical Phrase-based Machine Translation (SMT) systems have struggled with accuracy and the need for enormous data sets. Although Neural Machine Translation (NMT) is more accurate than previous methods like Rule-based MT and SMT, it still falls short of manual human translation. To the best of our knowledge, our multi-modal NMT receives the highest score for English to Hindi multi-modal translation. The ability of neural architectures to learn through machine learning is likely to improve, reducing the value provided by the features we studied.

In this paper, we apply NMT to one of the most difficult Indian-speaking pairs (English-Hindi). Data pre-processing and tokenization issues have been addressed. To cope with morphology and word issues in indigenous languages, we used Sequence to Sequence modelling techniques such as Encoding-Decoding, Recurrent Neural Network (RNN), and Long and Short Term Memory (LSTM) with 0.2, 0.9, and 1.0 BLEU values. We're also using a graphical user interface here (GUI). The same approach can be applied to other Indian languages. As our model is pretty accurate, it may be used to create translation software for English-Hindi, which is extremely valuable in areas such as tourism, education or business.

IX. ACKNOWLEDGEMENT

While bringing out this project to its final form, I came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey my gratitude to all of them. First and foremost, I would like to express my deep sense of gratitude and indebtedness to my supervisor Dr. R. Ravinder Reddy for his valuable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. I am also thankful to the Head of the Department Dr. Y. Rama Devi for providing excellent infrastructure and such a nice atmosphere for completing this project successfully. Finally, I would like to take this opportunity to thank my family and friends for their support throughout this work. I also sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

REFERENCES

[1] Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2017). Unsupervised neural machine translation. CoRR, abs/1710.11041. Booth, A. D. (1955). Machine translation of languages, fourteen essays.

[2] Cho, K., Van Merrienboer, B., Bahdanau, D., and Ben- ¨ gio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

[3] Fadaee, M., Bisazza, A., and Monz, C. (2017). Data augmentation for low-resource neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 567–573, Vancouver, Canada, July. Association for Computational Linguistics.

[4] Gage, P. (1994). A new algorithm for data compression. The C Users Journal, 12(2):23–38. Ghosh, S., Thamke, S., et al. (2014). Translation of telugumarathi and vice-versa using rule based machine translation. arXiv preprint arXiv:1406.3969.

[5] Hans, K. and Milton, R. (2016). Improving the performance of neural machine translation involving morphologically rich languages. arXiv preprint arXiv:1612.02482.

[6] Heinzerling, B. and Strube, M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In NicolettaCalzolari (Conference chair), et al., editors, Proceedings of the Eleventh
International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

[7] Jawaid, B. and Zeman, D. (2011). Word-order issues in English-to-Urdu statistical machine translation. Khan, N. J., Anwar, W., and Durrani, N. (2017). Machine translation approaches and survey for indian languages. arXiv preprint arXiv:1701.04290.

[8] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311– 318. Association for Computational Linguistics.

[9] Patel, R. N., Pimpale, P. B., et al. (2017). Mtil17: English to Indian langauge statistical machine translation. arXiv preprint arXiv:1708.07950. Patel, R., Pimpale, P., and Mukundan, S. (2018). Machine translation in indian languages: Challenges and resolution. Journal of Intelligent Systems, 09.

[10] Pathak, A. and Pakray, P.). Neural machine translation for indian languages. Journal of Intelligent Systems.

[11] Sandeep Saini, Vineet Sahula, in 2018 IEEE Fourth International Conference on Information Retrieval and Knowledge Management, in "Neural Machine Translation For English to Hindi".

[12] S.R.Laskar, A.Dutta, P.Pakray and S.Bandyopadhyay, in 2019 IEEE Conference on Information and Communication Technology, in "Neural Machine Translation: English to Hindi".

[13] S.P.Singh, H.Darbari, A.Kumar, S.Jain and A.Lohan, in 2008 IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), in "Overview of Neural Machine Translation for English-Hindi".

[14] S.Gogineni, G.Suryanarayana and S.K.Surendran, in IEEE International Conference on Smart Electronics and Communication (ICOSEC), in "An Effective Neural Machine Translation for English to Hindi".

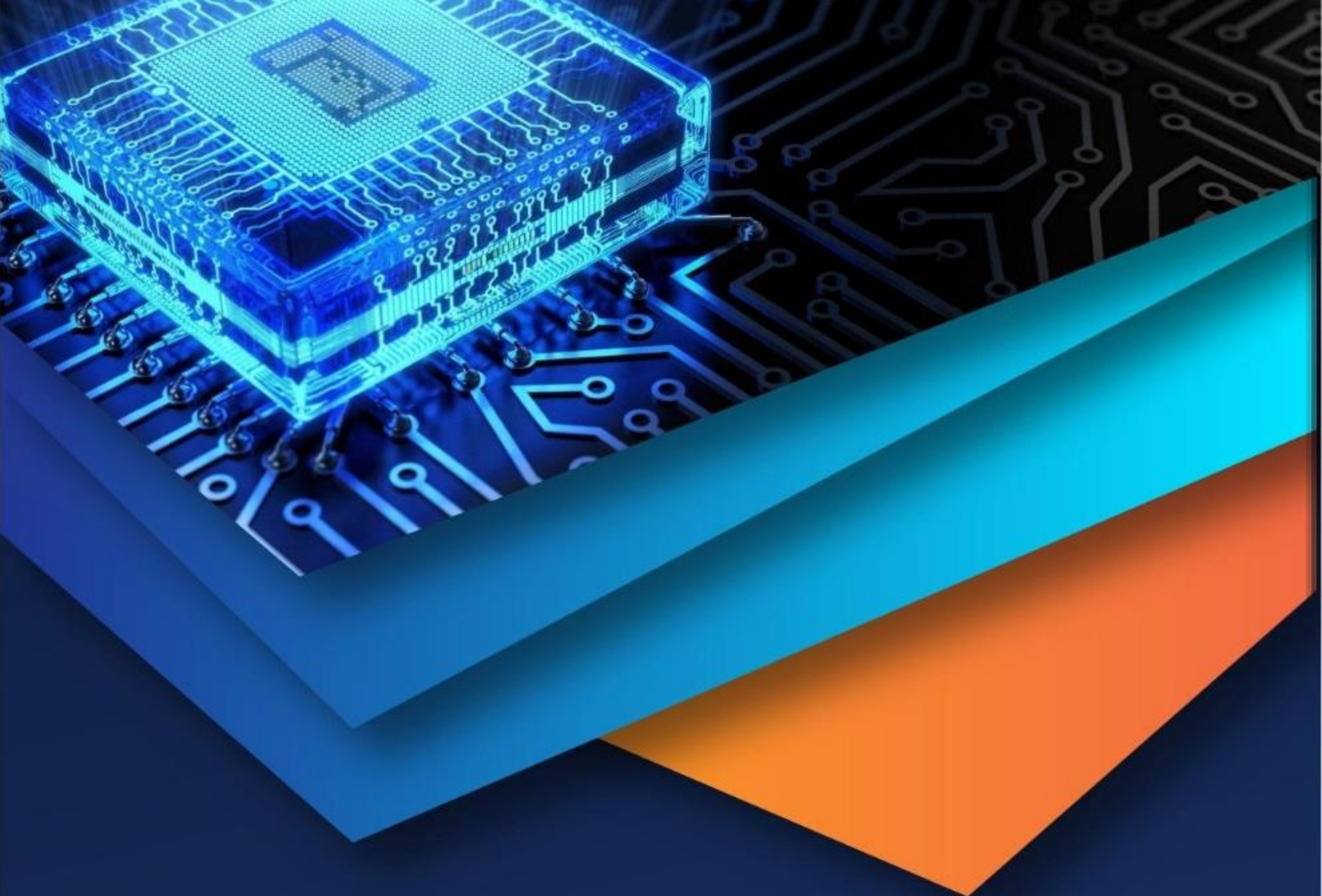
[15] B.Zhang, D.Xiong and J.Su, in 2018 IEEE Transactions on Pattern Analysis and Machine Intelligence, in "Neural Machine Translation with Deep Attention".

[16] N.Li, W.Su, Y.Li, H.Yu, W.Xu and B.Gao, in 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), in "Research on Machine Translation Automatic Evalution Based on Extended Reference".

[17] M.Singh, R.Kumar and I.Chana, in 2019 IEEE 7th International Conference on Smart Computing & Communications (ICSCC), in "Improving Neural Machine Translation using Rule based Machine Translation".

[18] P.Basmatkar, H.Holani and S.Kaushal, in 2019 IEEE 3rd International Conference on Computing Methodologies and Communication (ICCMC), in "Survey on Neural Machine Translation for Multilingual translation system".

[19] M. Singh, R. Kumar and I. Chana, in 2019 IEEE 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), in "Encoding-Decoding Methods for Neural Machine Translation".



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY