

LANGUAGE TRANSLATION - Transformer Based Attention Model

Submitted by

Ayushi Kumari-13000118110
Kumar Saurabh-13000118094
Abhijeet karmakar-13000118141
Abhay Kumar Choubey-13000118142

Submitted for the partial fulfillment for the degree of Bachelor of
Technology in Computer Science and Engineering



Techno Main Salt Lake,
EM 4/1, SaltLake, Sector V, Kolkata – 700 091

CERTIFICATE

This is to certify that the project entitled “Language Translation-Transformer based Attention Model” prepared by(Ayushi Kumari-13000118110 ,Kumar Saurabh-13000118094,Abhijeet karmakar-13000118141 ,Abhay Kumar Choubey-13000118142),.) of B.Tech (Computer Science & Engineering), Final Year, has been done according to the regulations of the Degree of Bachelor of Technology in Computer Science & Engineering. The candidates have fulfilled the requirements for the submission of the project report.

It is to be understood that the undersigned does not necessarily endorse any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

(Signature of the Internal Guide)

(Signature of the HOD)

(Signature of External Guide, if applicable)

(Signature of the External Examiner)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our mentor of the department of Computer Science and Engineering, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staff for the gracious hospitality they offered us.

Place: Techno Main Salt Lake

Date:

.....

.....

.....

.....

Contents

1	Introduction	5
	Abstract	5
	Problem Domain	5
	Related Studies	5
	Glossary	5
2	Problem Definition	7
	Scope 7	
	Exclusions	7
	Assumptions	7
3	Project Planning	8
	Software Life Cycle Model	8
	Scheduling	8
	Cost Analysis	10
4	Requirement Analysis	11
	Requirement Matrix	11
	Requirement Elaboration	12
5	Design	12
	Technical Environment	12
	Detailed Design	13
6	Implementation	17
	Implementation Details	17
	System Installation Steps	18
	System Usage Instructions	19
7	Test Results and Analysis	20
8	Conclusion	23
	Project Benefits	23
	Future Scope for improvements	24
9	References / Bibliography	25

1 Introduction

Abstract

The scientific community is rapidly generating various algorithms on Machine Learning that have helped in various fields of image processing, translation and many more. One of the most dominant work in the field of data science was proposed in 2017 which produced remarkable results. The research paper tells us about the implementation of transformer based attention model. Attention is a concept that helped improve the performance of neural machine translation applications. We are implementing this model to show the translation of language which can further be implemented for the protein sequence analysis.

Problem Domain

In this project, we are translating from Germany to English. Say the following sentence is an input sentence we want to translate: **"The animal didn't cross the street because it was too tired"**. What does "it" in this sentence refer to? Is it referring to the street or to the animal? It's a simple question to a human, but not as simple to an algorithm. In this project, we will identify this.

Related Studies

Through our study we came to know that, language translation needs proper algorithm to convert the sentences and by this it will help people we will be able to change sentences from one language to another with a proper meaning.

Glossary

Transformer:-A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV).

Encoder:- The encoder in the transformer consists of multiple encoder blocks. An input sentence goes through the encoder blocks, and the output of the last encoder block becomes the input features to the decoder.

Decoder:- The decoder also consists of multiple decoder blocks.

Self-Attention Models:-Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of a sequence.

2 Problem Definition

Scope

The project will accomplish the following objectives:

- Implementation of the transformer based attention model.
- Language Translation

Exclusions

We have 2 lakh training set, accuracy would be around 90percent.

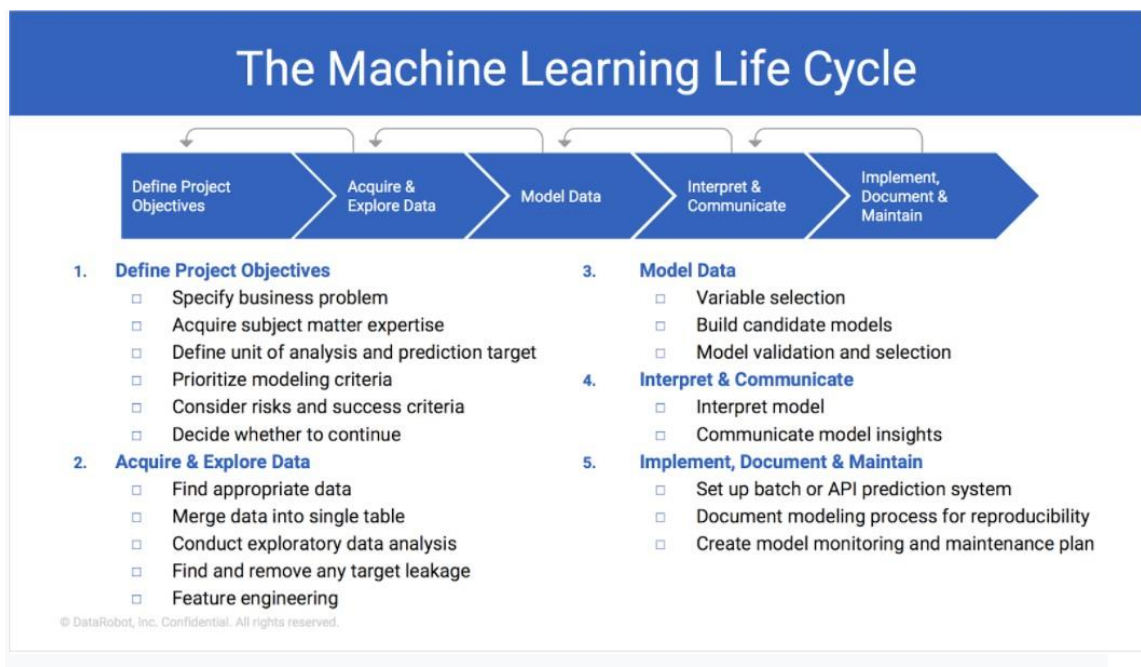
Assumptions

- The encoding component consists of 6 encoders stacked on top of each other.
- The decoding component is a stack of decoder of same number.
- The score is divided by 8 i.e square root of the dimension of the key vectors as it provides the most stable gradients.

3 Project Planning

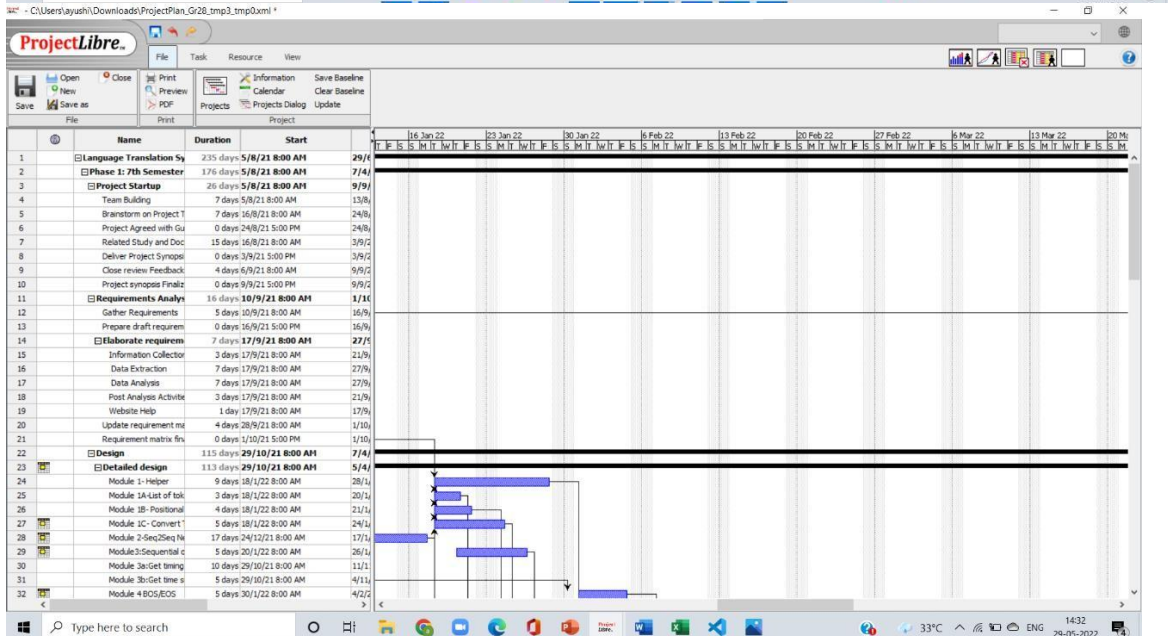
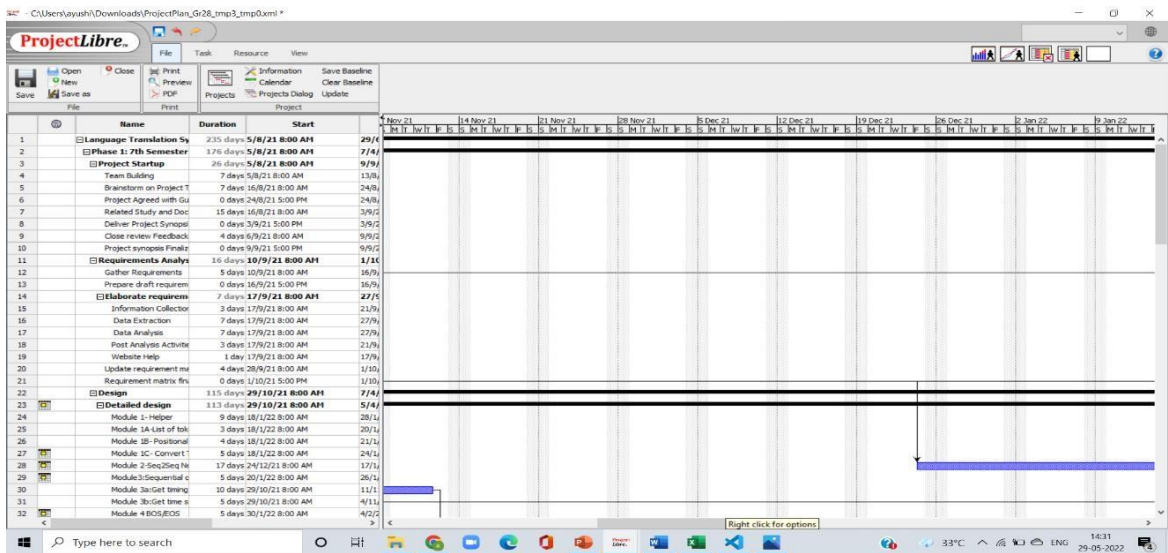
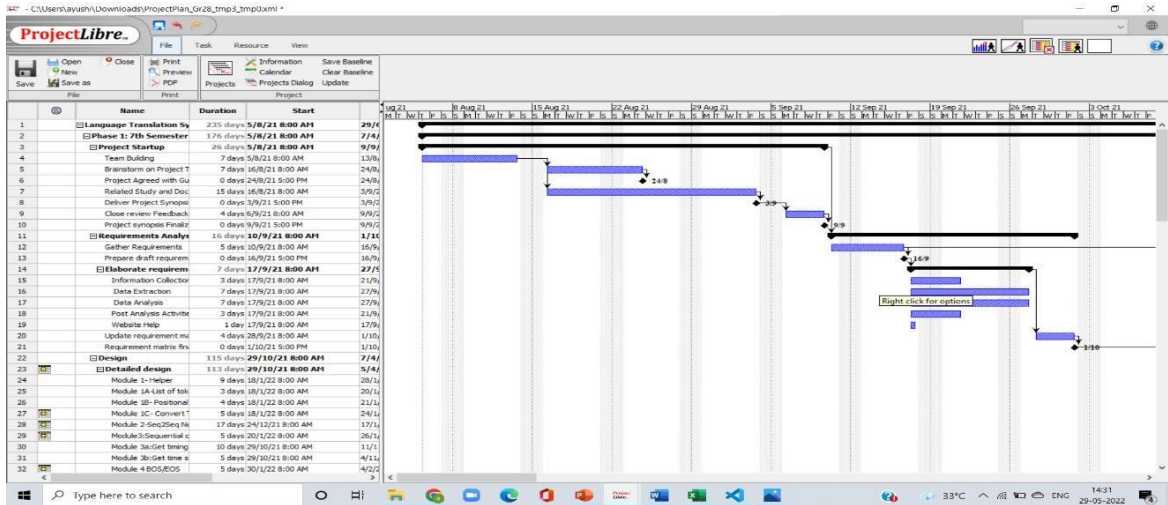
Software Life Cycle Model

There are five major steps in the machine learning life cycle, all of which have equal importance and go in a specific order.



Scheduling

LANGUAGE TRANSLATION -Transformer Based Attention Model



Cost Analysis

Our software is an application program. We are using Constructive Cost Model (COCOMO) for cost estimation.

Our project falls under the organic type with 42 kilo lines of code (KLOC).

We shall do the calculations using the basic model because the basic COCOMO is used in Organic mode by default.

Basic model:

The arithmetic formula of Basic COCOMO is:

a) Effort applied to the project: $E = a(KLOC)^b$ (in person-months)

b) Development time: $D = c(E)^d$ (in month)

where a, b, c, d are constraints for each category of software product.

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

so for our model a=2.4, b= 1.05, c= 2.5 and d=0.38.

Effort applied to the Project:

$$\begin{aligned}
 E &= a(KLOC)^b \\
 &= 2.4 * (42)^{1.05} \\
 &= 2.4 * 50.63 \\
 &= 121.512\text{-person month}
 \end{aligned}$$

LANGUAGE TRANSLATION -Transformer Based Attention Model

Development Time:

$$\begin{aligned} D &= c(E)^d \\ &= 2.5 * (121.512)^{0.38} \\ &= 2.5 * 6.20 \\ &= 15.5 \text{ months} \end{aligned}$$

4 Requirement Analysis

Requirement Matrix

RM_Requirements_Matrix_TSL_Template v1.6 (1) [Protected View] - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

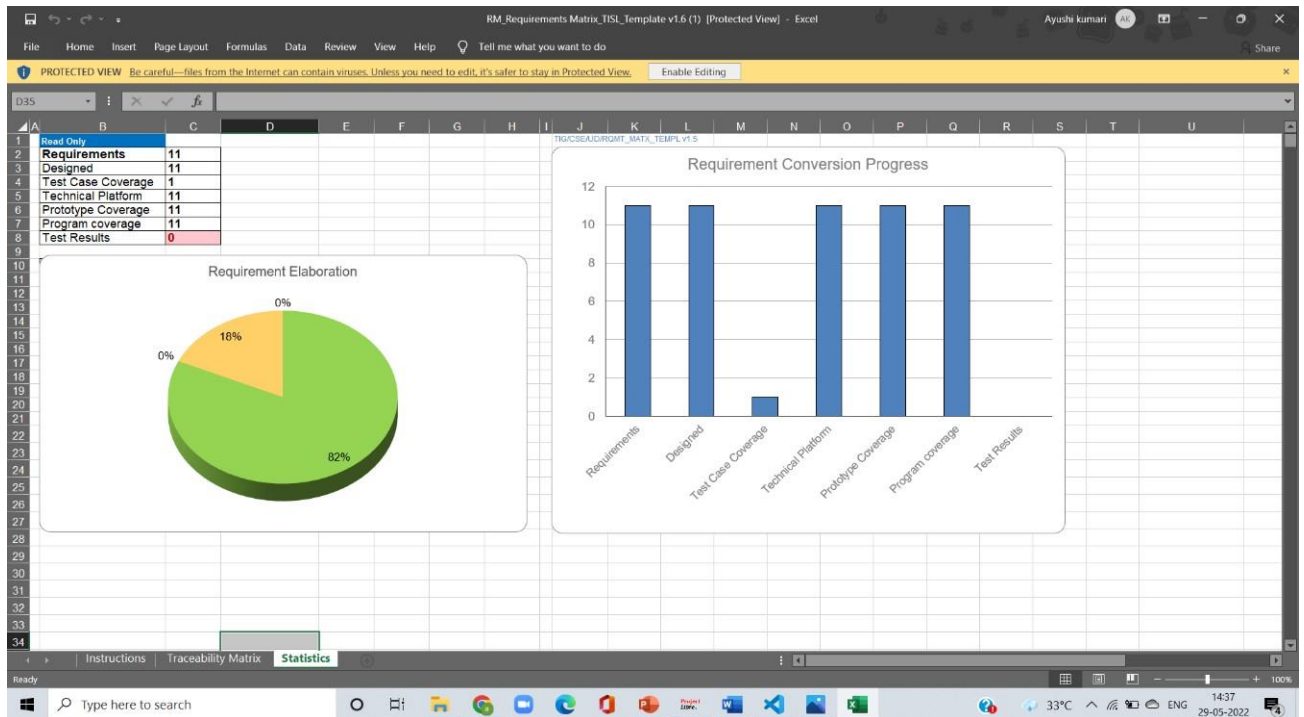
PROTECTED VIEW Be careful—files from the internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

	A	B	C	D	E	F	G	H	I	J	K	L
	Req ID	Requirement Item	Requirement Analysis Status	Design Module	Design Reference (section/ sub project Report)	Test Case Number	Technical Platform or Implementation	Prototype prepared ?	Name of Program / Component	Test Results Reference	Additional Comments (if not included in previous columns)	
1	LT-1	Language Translation	In-progress	LT	5.2	T4.GN-1	Python	Yes	translate.py			
2		helper function to yield list of tokens	Completed	LT	5.2.1		Python	Yes	yield_tokens(data_loader)			
3		helper Module that adds positional encoding to the token embedding to introduce a	Completed	LT	5.2.2		pytorch torchtext.spacy	Yes	PositionalEncoding(module)			
4		helper Module to convert tensor of input indices into corresponding tensor of tokens	Completed	LT	5.2.3		pytorch torchtext.spacy	Yes	TokenEmbedding(module)			
5		Seq2Seq Network	Completed	LT	5.2.4		pytorch torchtext.spacy	Yes	Seq2SeqTransformer(module)			
6		helper function to club together sequential operations	Completed	LT	5.2.5		pytorch torchtext.spacy	Yes	sequential_transform("transform")			
7		function to add BOS/EOS and create tensor for input sequence indices	Completed	LT	5.2.6		pytorch torchtext.spacy	Yes	tensor_transform(token_id_collate_fn(batch))			
8		function to collate data samples into batch tensors	Completed	LT	5.2.7		pytorch torchtext.spacy	Yes				
9		train and evaluate model	In-progress	LT	5.2.8		pytorch torchtext.spacy	Yes	train_epoch(model)			
10		function to generate output sequence using greedy algorithm	Completed	LT	5.2.9		pytorch torchtext.spacy	Yes	greedy_decode			
11		actual function to translate input sentence to target language	Completed	LT	5.2.10		pytorch torchtext.spacy	Yes	translate			
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												

Instructions Traceability Matrix Statistics

Ready

Requirement Elaboration



5 Design

Technical Environment

Hardware:

20 gm RAM

256 SSD

10 GB Graphic Card

Operating System:

LANGUAGE TRANSLATION -Transformer Based Attention Model

Ubuntu

Software:-

Text Editor

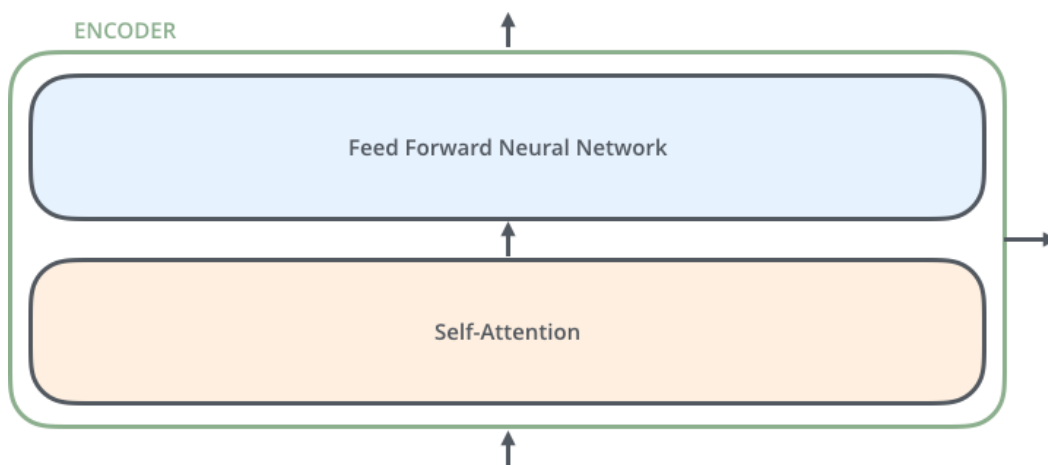
Spacy

TorchText

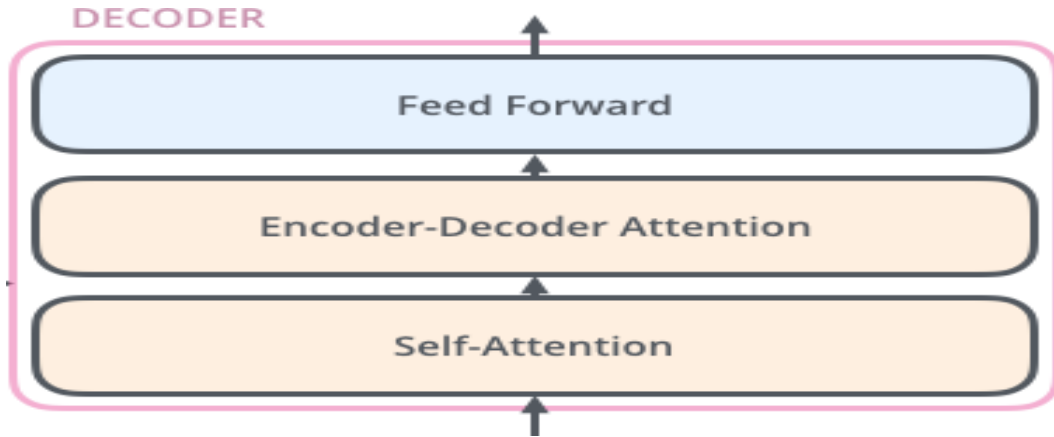
pyTorch

Detailed Design

Encoder:-

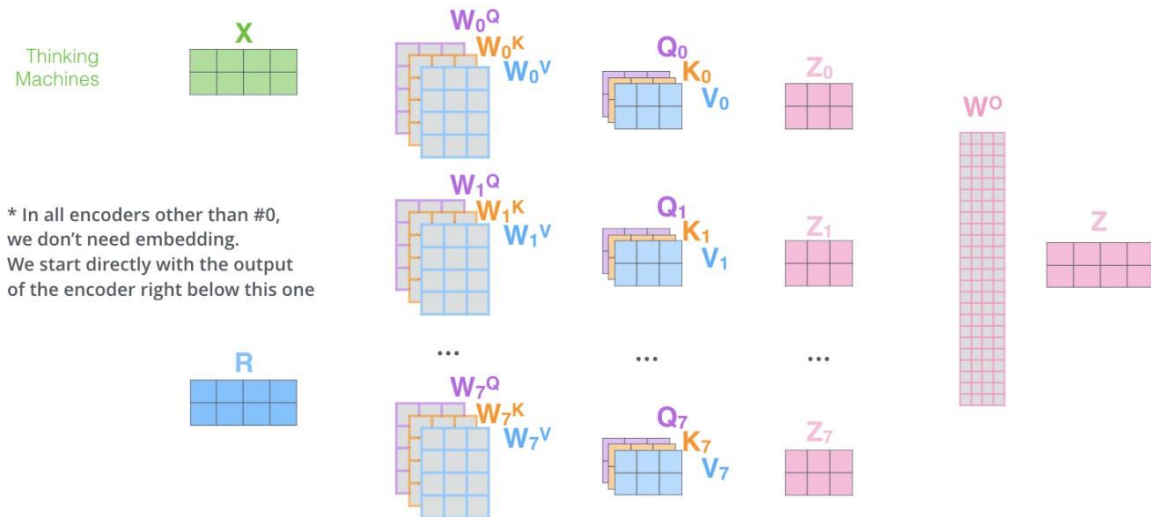


Decoder:-



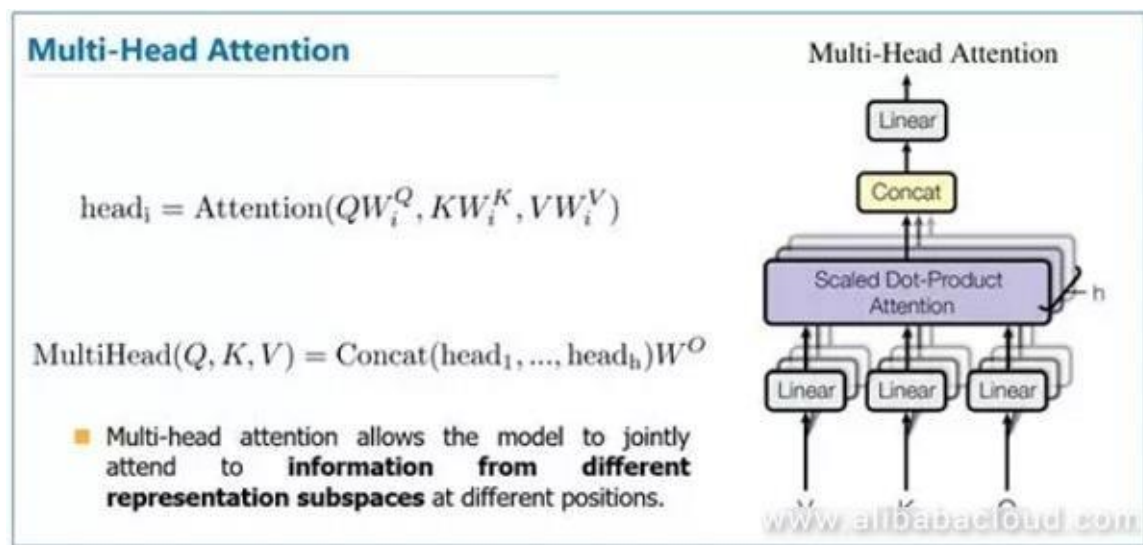
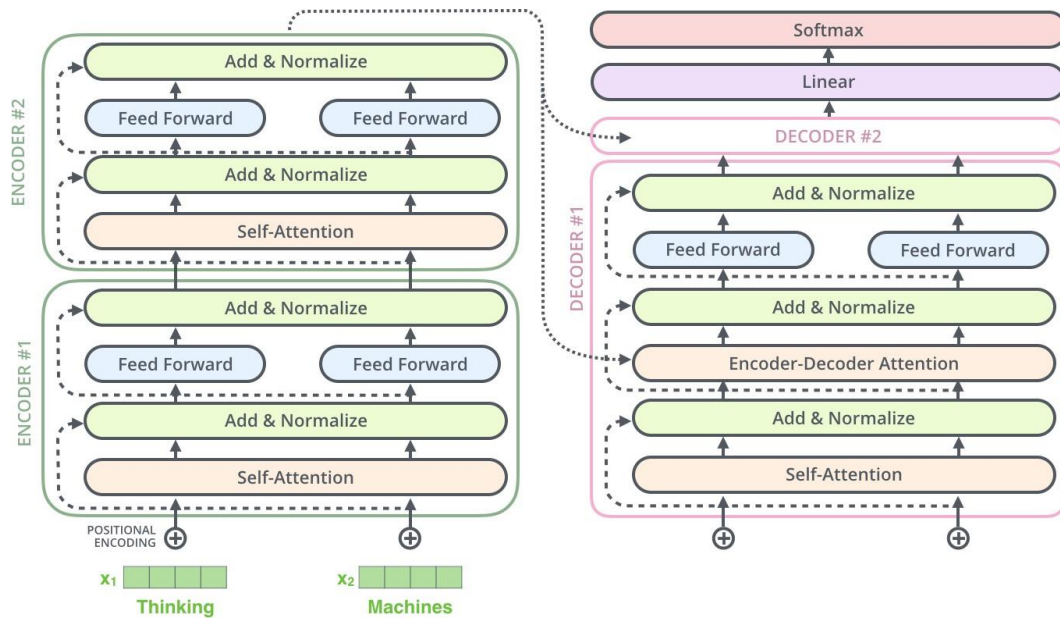
Multi-headed self-attention

- 1) This is our input sentence*
- 2) We embed each word*
- 3) Split into 8 heads. We multiply X or R with weight matrices
- 4) Calculate attention using the resulting $Q/K/V$ matrices
- 5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



Overview of encoder and decoder

LANGUAGE TRANSLATION -Transformer Based Attention Model



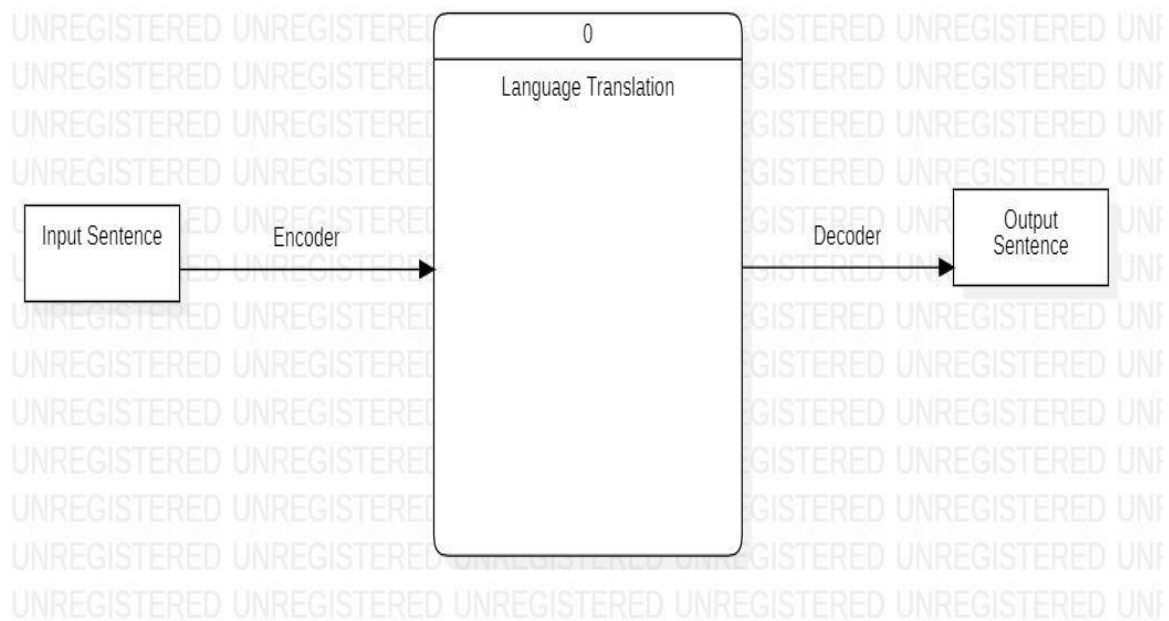
Calculation of SoftMax

LANGUAGE TRANSLATION -Transformer Based Attention Model

$$\text{softmax}\left(\frac{\begin{matrix} \text{Q} \\ \begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix} \times \begin{matrix} \text{K}^T \\ \begin{matrix} \square & \square \\ \square & \square \end{matrix} \end{matrix}}{\sqrt{d_k}}\right) \begin{matrix} \text{V} \\ \begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix}$$

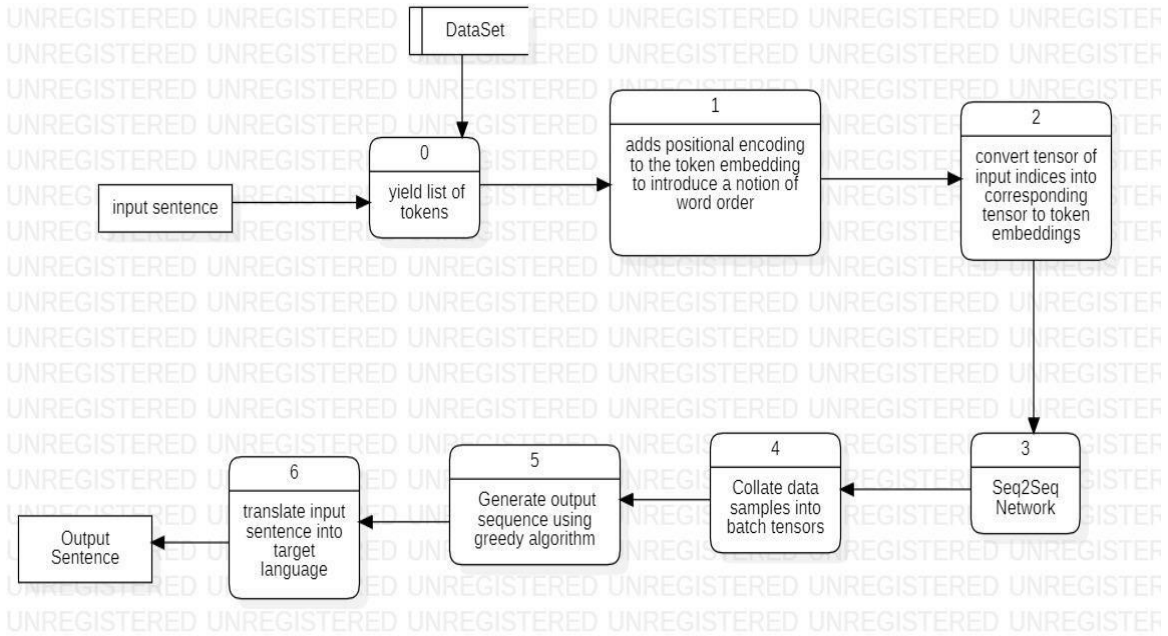
$$= \begin{matrix} \text{Z} \\ \begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix}$$

DFD Level-0



DFD Level-1

LANGUAGE TRANSLATION -Transformer Based Attention Model



6 Implementation

Implementation Details

- Module to yield list of tokens
- Module that adds positional encoding to the token embedding to introduce a notion of word order.
- Module to convert tensor of input indices into corresponding tensor to token embeddings.
- Seq2Seq Network
- Collate data samples into batch tensors
- Generate output sequence using greedy algorithm

- g. Function to translate input sentence into target language
- h. Source and target language text transforms to convert raw strings to tensors indices

System Installation Steps

We have installed four software for running the project:-

1. Python

Commands :-

Sudo apt update

Sudo apt install software-properties-common

Sudo apt install python3.8

2. Spacy

Commands:-

Pip install -U pip setuptools wheel

Pip install -U spacy

Pip -m spacy download en core wen sm

3. Torchtext

Commands:-

Pip install torchtext

4. PyTorch

LANGUAGE TRANSLATION -Transformer Based Attention Model

Commands:-

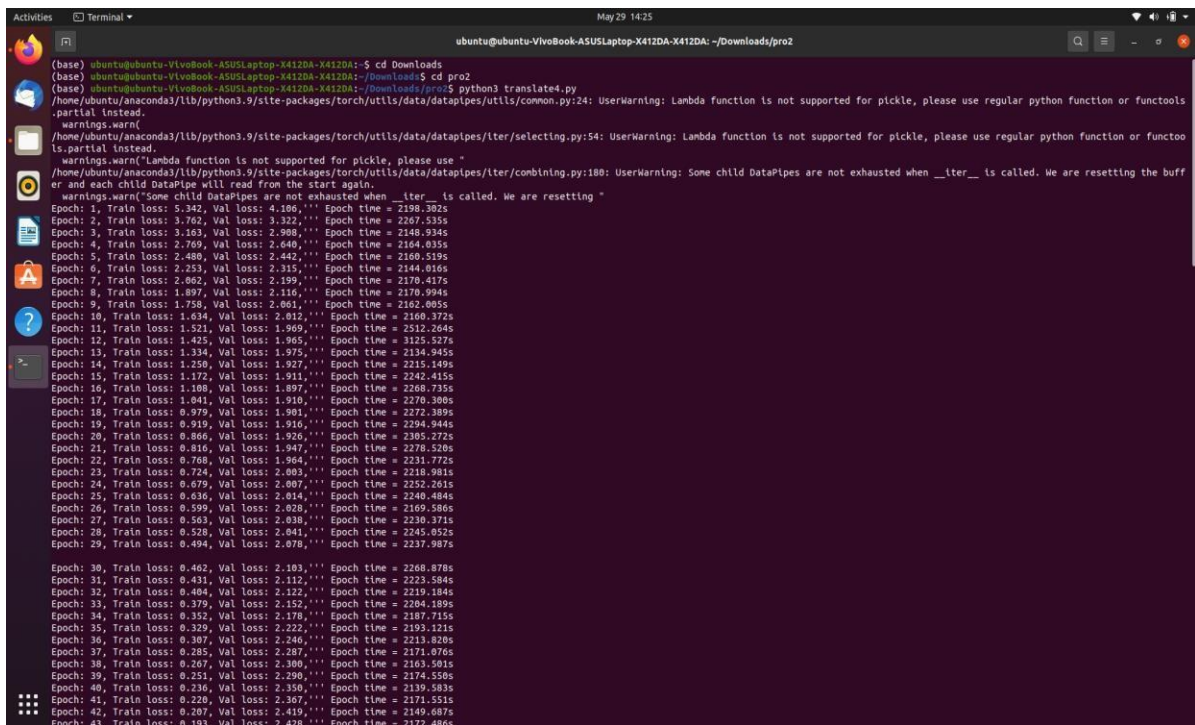
Sudo apt install python3-pip

Pip3 install torch==1.9+cpu torchvision==0.10.1+cpu-f

https://download.pytorch.org/whl/torch_stable.html

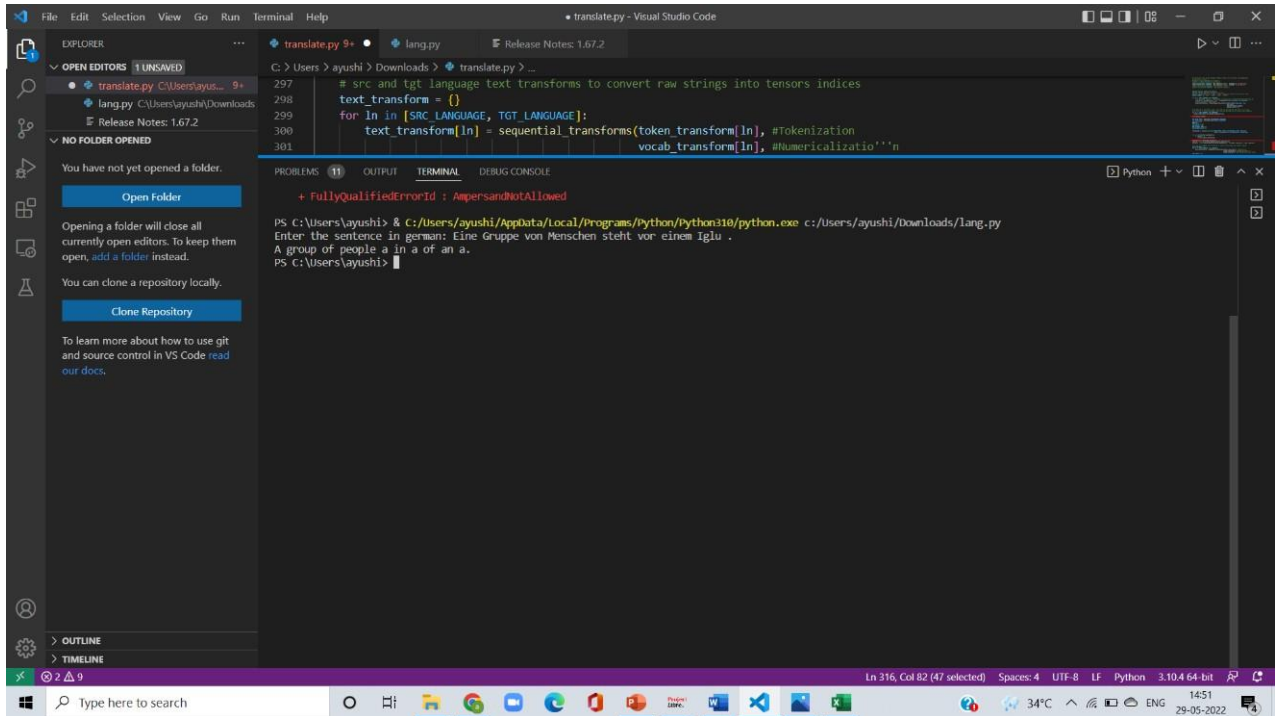
System Usage Instructions

Command to run the application:-



```
(base) ubuntu@ubuntu-VivoBook-ASUSLaptop-X412DA-X412DA:~$ cd Downloads
(base) ubuntu@ubuntu-VivoBook-ASUSLaptop-X412DA-X412DA:~/Downloads$ cd pro2
(base) ubuntu@ubuntu-VivoBook-ASUSLaptop-X412DA-X412DA:~/Downloads/pro2$ python3 translate4.py
/home/ubuntu/anaconda3/lib/python3.9/site-packages/torch/utils/data/datapipe/utils/common.py:24: UserWarning: Lambda function is not supported for pickle, please use regular python function or functools.partial instead.
  warnings.warn(
/home/ubuntu/anaconda3/lib/python3.9/site-packages/torch/utils/data/datapipe/iter/selecting.py:54: UserWarning: Lambda function is not supported for pickle, please use regular python function or functools.partial instead.
  warnings.warn(
/home/ubuntu/anaconda3/lib/python3.9/site-packages/torch/utils/data/datapipe/iter/combining.py:180: UserWarning: Some child DataPipes are not exhausted when __iter__ is called. We are resetting the buffer and each child DataPipe will read from the start again.
  warnings.warn("Some child DataPipes are not exhausted when __iter__ is called. We are resetting "
Epoch: 1, Train loss: 5.342, Val loss: 4.106, Epoch time = 2198.301s
Epoch: 2, Train loss: 3.762, Val loss: 3.322, Epoch time = 2267.535s
Epoch: 3, Train loss: 3.163, Val loss: 2.988, Epoch time = 2148.934s
Epoch: 4, Train loss: 2.769, Val loss: 2.648, Epoch time = 2164.035s
Epoch: 5, Train loss: 2.480, Val loss: 2.442, Epoch time = 2160.519s
Epoch: 6, Train loss: 2.253, Val loss: 2.315, Epoch time = 2144.016s
Epoch: 7, Train loss: 2.062, Val loss: 2.199, Epoch time = 2170.417s
Epoch: 8, Train loss: 1.897, Val loss: 2.116, Epoch time = 2170.994s
Epoch: 9, Train loss: 1.750, Val loss: 2.061, Epoch time = 2162.005s
Epoch: 10, Train loss: 1.634, Val loss: 2.012, Epoch time = 2160.372s
Epoch: 11, Train loss: 1.521, Val loss: 1.969, Epoch time = 2512.264s
Epoch: 12, Train loss: 1.425, Val loss: 1.965, Epoch time = 3125.527s
Epoch: 13, Train loss: 1.334, Val loss: 1.975, Epoch time = 2134.945s
Epoch: 14, Train loss: 1.250, Val loss: 1.927, Epoch time = 2215.149s
Epoch: 15, Train loss: 1.172, Val loss: 1.911, Epoch time = 2242.415s
Epoch: 16, Train loss: 1.100, Val loss: 1.897, Epoch time = 2268.735s
Epoch: 17, Train loss: 1.041, Val loss: 1.910, Epoch time = 2270.300s
Epoch: 18, Train loss: 0.979, Val loss: 1.901, Epoch time = 2272.389s
Epoch: 19, Train loss: 0.919, Val loss: 1.916, Epoch time = 2294.944s
Epoch: 20, Train loss: 0.866, Val loss: 1.926, Epoch time = 2305.272s
Epoch: 21, Train loss: 0.816, Val loss: 1.947, Epoch time = 2276.520s
Epoch: 22, Train loss: 0.768, Val loss: 1.964, Epoch time = 2231.772s
Epoch: 23, Train loss: 0.724, Val loss: 2.003, Epoch time = 2218.981s
Epoch: 24, Train loss: 0.679, Val loss: 2.007, Epoch time = 2252.261s
Epoch: 25, Train loss: 0.636, Val loss: 2.014, Epoch time = 2240.404s
Epoch: 26, Train loss: 0.599, Val loss: 2.028, Epoch time = 2169.586s
Epoch: 27, Train loss: 0.563, Val loss: 2.038, Epoch time = 2230.371s
Epoch: 28, Train loss: 0.528, Val loss: 2.041, Epoch time = 2245.052s
Epoch: 29, Train loss: 0.494, Val loss: 2.078, Epoch time = 2237.907s
Epoch: 30, Train loss: 0.462, Val loss: 2.103, Epoch time = 2268.878s
Epoch: 31, Train loss: 0.431, Val loss: 2.112, Epoch time = 2223.584s
Epoch: 32, Train loss: 0.404, Val loss: 2.122, Epoch time = 2219.184s
Epoch: 33, Train loss: 0.379, Val loss: 2.152, Epoch time = 2204.189s
Epoch: 34, Train loss: 0.352, Val loss: 2.178, Epoch time = 2187.715s
Epoch: 35, Train loss: 0.329, Val loss: 2.222, Epoch time = 2193.121s
Epoch: 36, Train loss: 0.307, Val loss: 2.246, Epoch time = 2213.820s
Epoch: 37, Train loss: 0.285, Val loss: 2.287, Epoch time = 2171.876s
Epoch: 38, Train loss: 0.267, Val loss: 2.300, Epoch time = 2163.501s
Epoch: 39, Train loss: 0.251, Val loss: 2.290, Epoch time = 2174.550s
Epoch: 40, Train loss: 0.236, Val loss: 2.350, Epoch time = 2139.583s
Epoch: 41, Train loss: 0.220, Val loss: 2.367, Epoch time = 2171.551s
Epoch: 42, Train loss: 0.207, Val loss: 2.419, Epoch time = 2149.687s
Epoch: 43, Train loss: 0.191, Val loss: 2.426, Epoch time = 2172.486s
```

7 Test Results and Analysis



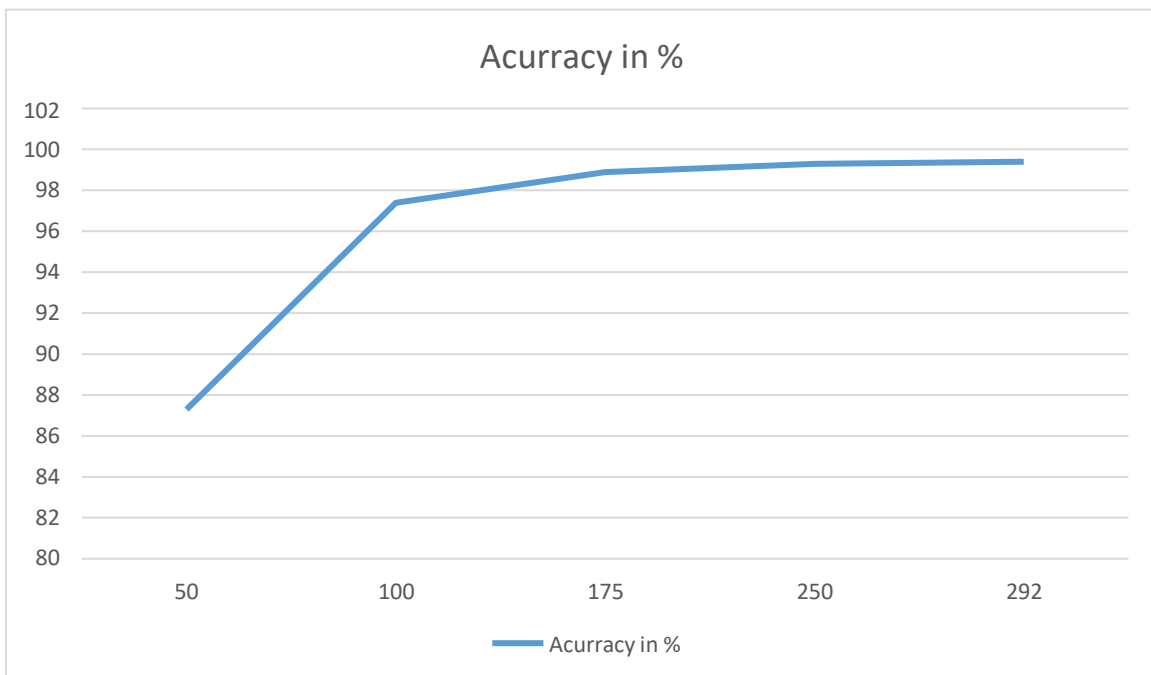
Serial No.	Test Cases	Output Result	Expected Result	Status
1.	Eine Gruppe von Menschen stehtvoreinem Iglu.	A group of people a in a of an a.	A group ofpeople stand in front of an igloo.	40 percent
2.	Es istunserAbschlussprojekt	It's our final a project	It's our final Year project	50 percent
3.	Unser ProjektnameistSprachübersetzung	Our project a	Our project name is	40 percent

LANGUAGE TRANSLATION -Transformer Based Attention Model

		Language a	language translation	
--	--	------------	----------------------	--

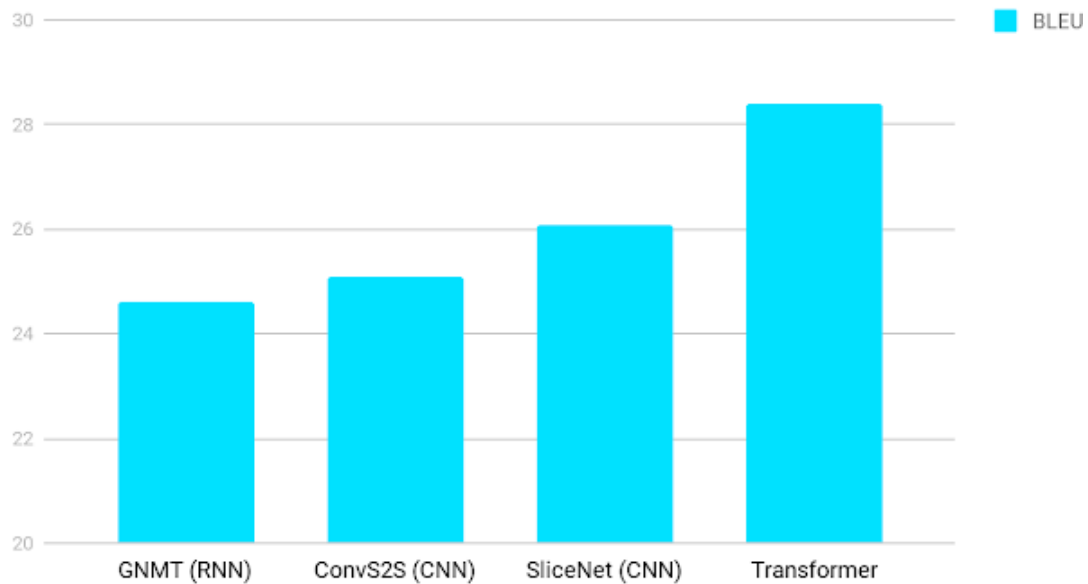
Performance Metrics:

SL.NO.	Epoch size	Training loss	Validation Loss	Accuracy
1	50	0.127	2.521	87.3%
2	100	0.026	3.193	97.4%
3	175	0.011	3.541	98.9%
4	250	0.007	3.733	99.3%
5	292	0.006	3.792	99.4%



Comparative study:

English German Translation quality



Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Analysis:-

The test cases is almost 40%accuarte.

It depends upon how many epochs we are running .More epochs, more learning will be there and more accurate result will come.

8 Conclusion

Project Benefits

The ability to communicate with one another is a fundamental part of being human. There are nearly 7,000 different languages worldwide. As our world becomes increasingly connected, language translation provides a critical cultural and economic bridge between people from different countries and ethnic groups. Some of the more obvious use-cases include:

- business: international trade, investment, contracts, finance
- commerce: travel, purchase of foreign goods and services, customer support
- media: accessing information via search, sharing information via social networks, localization of content and advertising
- education: sharing of ideas, collaboration, translation of research papers
- government: foreign relations, negotiation

To meet these needs, technology companies are investing heavily in machine translation. This investment and recent advancements in deep learning have yielded major improvements in translation quality. According to Google, switching to deep learning produced a 60% increase in translation accuracy compared to the phrase-based approach previously used in Google Translate. Today, Google and Microsoft can translate over 100 different languages and are approaching human-level accuracy for many of them.

Future Scope for improvements

As transformer based model can be used in sequential based data it can be used in protein sequence analysis and in DNA sequence analysis

9 References / Bibliography

1. Alammar, Jay. “The Illustrated Transformer – Jay Alammar – Visualizing Machine Learning One Concept at a Time.” *Jay Alammar – Visualizing Machine Learning One Concept at a Time.*, <https://jalammar.github.io/illustrated-transformer/>. Accessed 2 June 2022.
2. Gaffar, Syed Abdul. “Language Translation with Transformer in Python!” *Analytics Vidhya*, 12 June 2021, <https://www.analyticsvidhya.com/blog/2021/06/language-translation-with-transformer-in-python/>.
3. Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.
4. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.
5. Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. CoRR, abs/1606.04199, 2016

