ANLP – Assignment 3

Course Coordinator: Manish Srivatsava
Name : Lakshmipathi Balaji
Roll No: 2021114007
Mail : lakshmipathi.balaji@gmail.com

Question:

What is the purpose of self-attention, and how does it facilitate capturing dependencies in sequences?

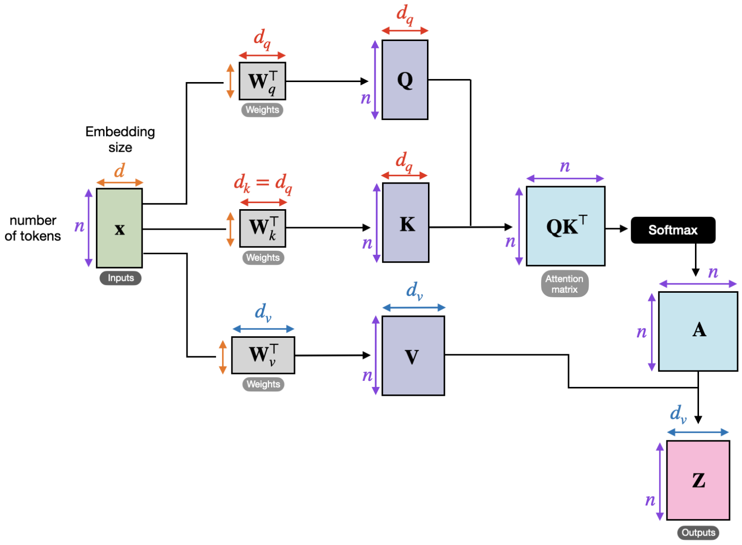
What is Self-Attention?

Self-attention is a mechanism used in machine learning (especially natural language processing (NLP) and computer vision tasks) that specializes in capturing dependencies and relationships within input sequences. It enables the model to identify and assign importance to different parts of the input sequence by attending to itself.

How Self-Attention Works

Transforming Input: Self-attention begins by transforming the input sequence into three vectors: query, key, and value. These vectors are produced through linear transformations of the input.

Weighted Sum Calculation: The attention mechanism calculates a weighted sum of these values based on the similarity between the query and key vectors.



Output Generation: The resulting weighted sum, together with the original input, is then passed through a feed-forward neural network to produce the final output. This process enables the model to focus on relevant information and captures long-range dependencies.

Importance of Self-Attention

Long-range Dependencies Resolution: It enables the model to capture relationships between distant elements in a sequence, empowering it to decipher complex patterns and dependencies.

Contextual Understanding: By focusing on different parts of the input sequence, the model understands the context better and assigns appropriate weights to each element based on its relevance.
Parallel Computation: Self-attention computations can execute in parallel for each element in the sequence, making it computationally efficient and scalable for large datasets.

Question:
Why do transformers use positional encodings in addition to word embeddings? Explain how positional encodings are incorporated into the transformer architecture

Importance of Positional Encodings in Transformers

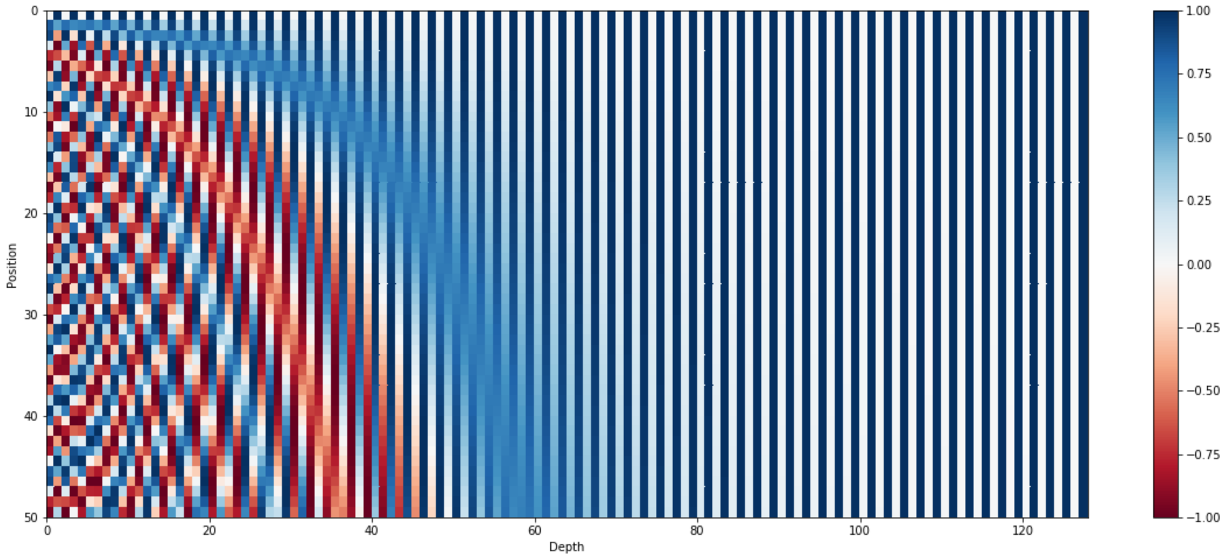
1. Role of Word Order in Language: The position and order of words are critical parts of any language. They define the grammar and semantics of a sentence. While Recurrent Neural Networks (RNNs) inherently take the order of words into account by parsing a sentence word by word in a sequential manner, Transformers follow a unique approach.

2. Deviation from Recurrence Mechanism: The Transformer architecture replaces the recurrence mechanism with a multi-head self-attention mechanism. Owing to the simultaneous flow of all the words in a sentence through the Transformer's encoder/decoder stack, the model lacks any inherent sense of word order. Therefore, there is a need for a methodology to incorporate word order information into the Transformers.

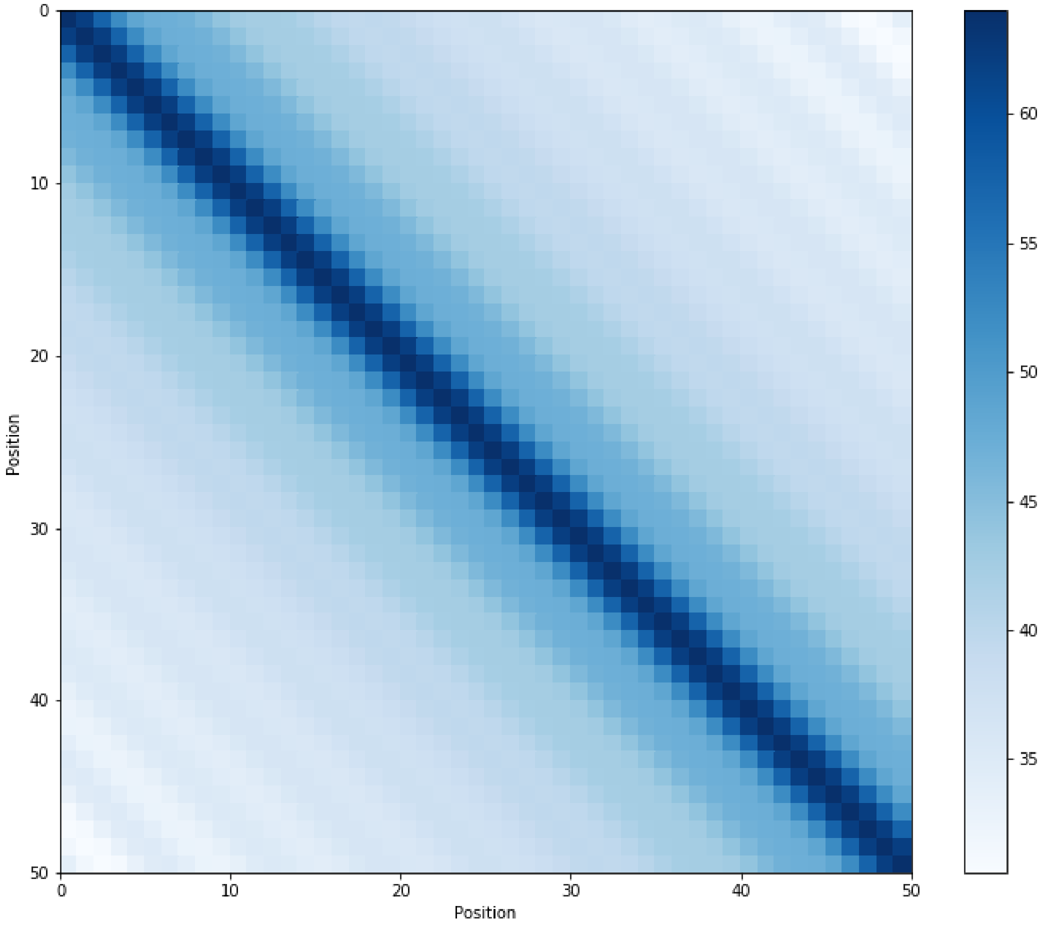
Incorporation of Positional Encodings in Transformers

3. Concept of Positional Encoding: The solution to imbue the model with a perception of order is to supplement every word with information about its position in the sentence. This additional piece of information is what we refer to as the positional encoding. The encoding is not a single number but a multi-dimensional vector that contains information about a specific position in a sentence.

4. Criteria for Positional Encoding: There are several criteria a suitable positional encoding should satisfy. Firstly, it should output a unique encoding for each time-step or word's position in a sequence. Secondly, the distance between any two time-steps should be consistent across sentences with varying lengths. Moreover, it should enable the model to generalize to longer sentences without any extra efforts, and its values should be bounded. Lastly, the generation of positional encoding must be deterministic.



Another property of sinusoidal position encoding is that the distance between neighboring time-steps are symmetrical and decays nicely with time.



5. <u>Sinusoidal Function as Positional Encoding</u>: Transformers meet these requirements by adopting a sinusoidal function for positional encoding. It injects the order of words and facilitates attendance by relative positions. This encoding forms a geometric progression from high to low frequencies along the vector dimension.

6. <u>Application to Word Embeddings</u>: In the original work, the positional encoding is added to the actual word embeddings. For every word in a sentence, the corresponding embedding which is fed to the model is calculated by adding its positional encoding to the word embedding. Such an approach allows the retention of positional information.

7. <u>Relative Positioning</u>: The sinusoidal positional encoding encourages the model to easily discern relative positions. That's because, for any fixed

offset, the positional encoding for position `pos + k` can be expressed as a linear function of the positional encoding for position `pos`. It helps the model to easily learn and attend to relative positions.

8.Handling Multitude of Layers: The problem of positional information fading as it reaches the upper layers is addressed via residual connections in the Transformer architecture. This enables efficient propagation of input information, which contains positional embeddings, to further layers where complex interactions are dealt with.

In conclusion, while word embeddings provide the model with an understanding of each word's content, positional encodings provide the model with an understanding of each word's context within a sentence. The combination of these two elements aids in the model's comprehension and construction of meaningful outputs.

REPORT

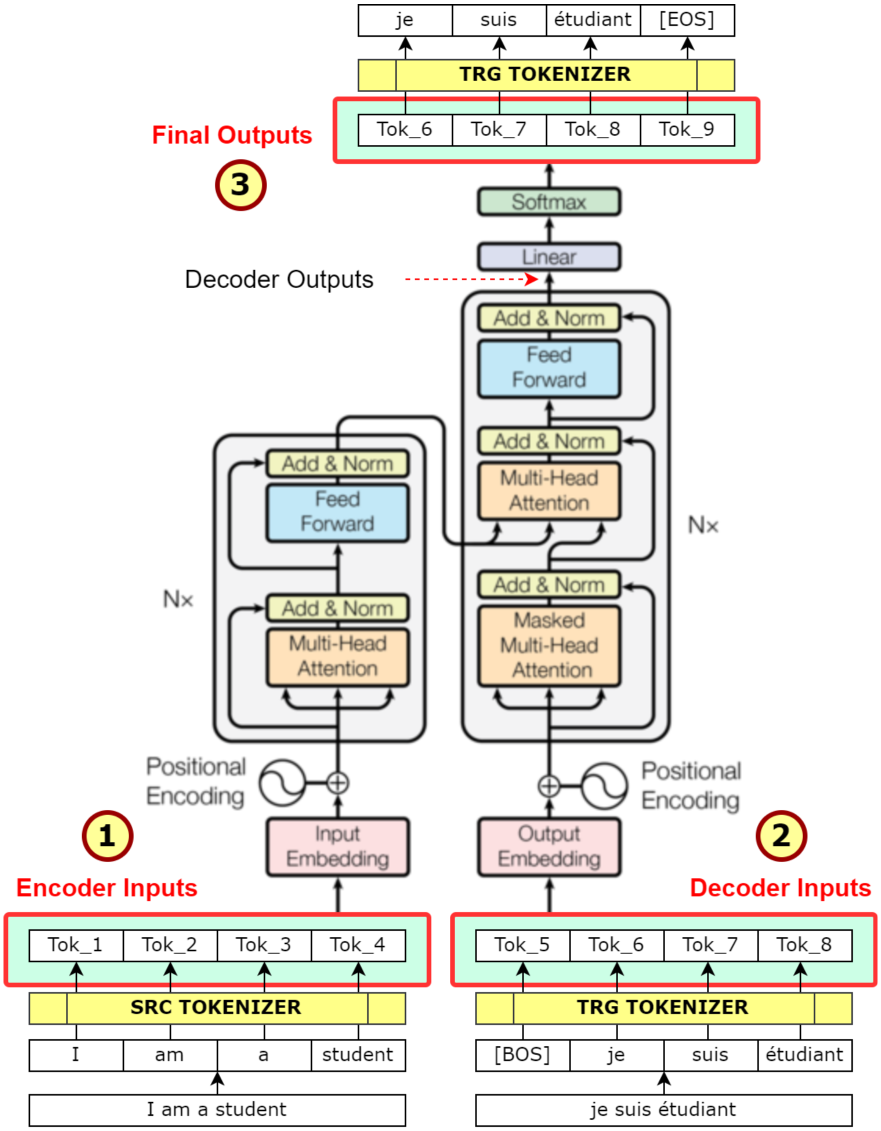
The dataset is a subset of the dataset
for the English-French translation task in IWSLT 2016. It contains the files for
train, dev and test splits as below:

1. train.[en—fr]: 30000 lines each
2. dev.[en—fr]: 887 lines each
3. test.[en—fr]: 1305 lines each

Parameter	Value
d_model	[256,512]
dropout	0.1
num_layers	1
optimizer	adam
batch_size	64
random_seed	42
n_heads	[2,4]
Metric	Bleu_score with n = 1, 2, 3

These are the parameters used for the task in a wanb sweep for the transformer class built with the help of paper references -

1. Attention is All You Need
2. The Illustrated Transformer, Jay Alammar



Though there other architectures to this like normalizing before layer(pre-norm), after layers(post-norm) the one suggested by paper is implemented, and teacher forcing method is followed for the training objective.

Teacher forcing method:

Teacher forcing is a training technique in sequence-to-sequence models where, during training, the model is provided with the ground truth (true target) tokens at each time step, rather than its own predictions. This helps stabilize training and accelerates convergence. However, it may lead to suboptimal performance during inference, as the model may not be robust to prediction errors.

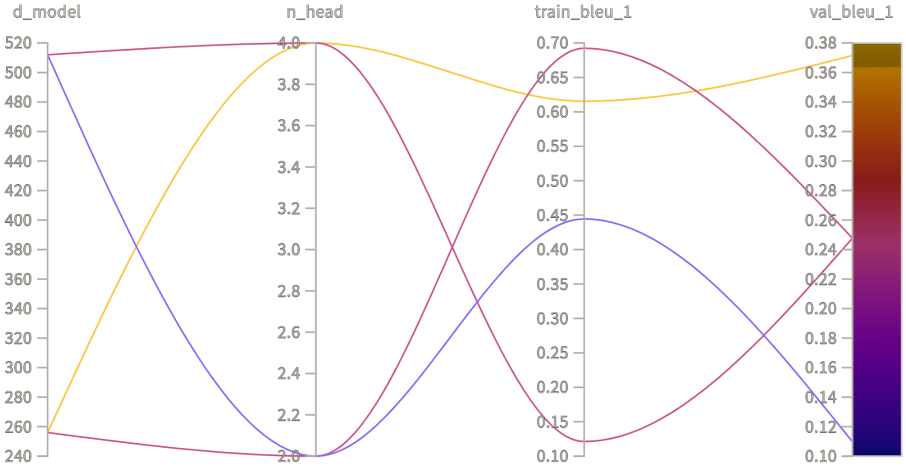
Bleu Score:

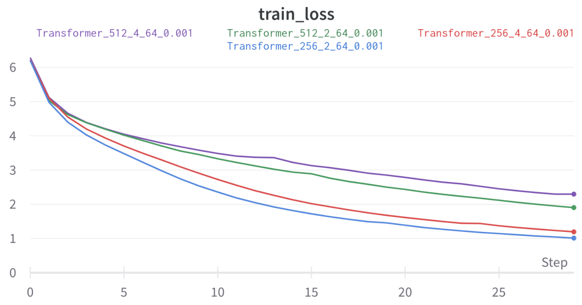
BLEU (Bilingual Evaluation Understudy) is a metric for evaluating the quality of machine-generated text, often used in machine translation tasks. It measures the similarity between generated text and reference text by comparing n-grams (contiguous word sequences) and computing a score between 0 and 1. Higher BLEU scores indicate better text generation quality.

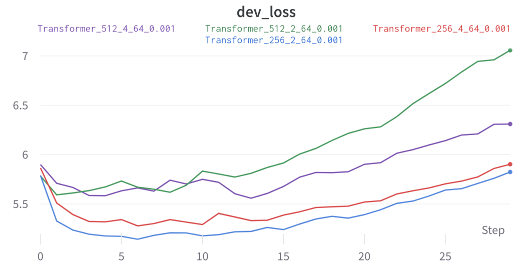
Note:

- NLTK punkt tokenizer is used for preprocessing.
- Sentence bleu metric from NLTK is used to calculate bleu score.
- Val_bleu_1_gram is used as heuristic to decide the best model.
- ROP scheduler is used for training.

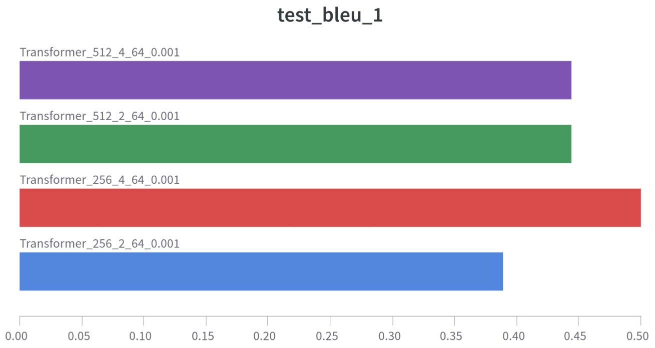
Here are some self-explainable diagrams based up on results, losses and other metrics across all combinations.

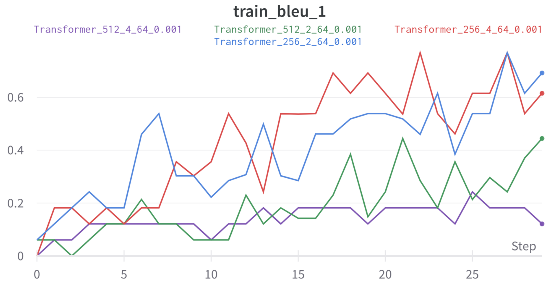


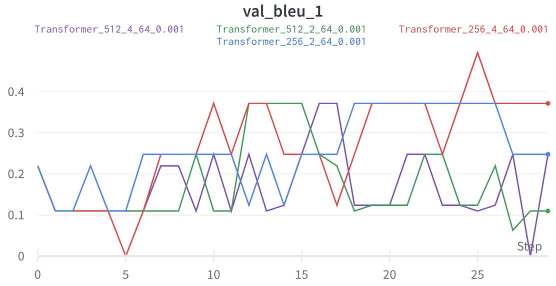




Note that the models are easily overfitted after a few epochs as the data is only 30,000 sentences which is considered as a small dataset for the usage of models like transformers as they are highly initialization dependent.







On comparing the val_metrics we can say that the model with *d_ model* as 256 and 4 *attention heads* is best for the given task within the hyperparameter combinations tried.

Conclusion:

In this assignment, I learned about transformer architecture thoroughly from scratch. The assignment also helps in better understanding of teacher forcing method and positional encoding.

Link to wandb project - wandb

Link to the models and files - ass3