

Section 1.5 Transformer-based Machine Translation (25')

```
=====
Computing BLEU score on full dataset...
```

```
BLEU score: 0.6074
=====
```

```
Saving model...
```

```
Model saved to 'transformer_translation_model.pt'
```

```
Training complete!
```

Section 2.1 - Transformer based Sentiment Classification

My hyperparameters using glove.6B.50d

- Num_encoder_layers = 3
- Embedding dimension 50
- nhead=2
- Optimizer: Adam
- num_epochs=50
- batch_size=48
- Droup_out = 0.1
- Learning_rate = 0.001

Model architecture

```
Transformer Model Architecture:
```

```
TransformerSentimentClassifier(
  (embedding): Embedding(16581, 50)
  (pos_encoder): PositionalEncoding(
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (transformer_encoder): TransformerEncoder(
    (layers): ModuleList(
      (0-2): 3 x TransformerEncoderLayer(
        (self_attn): MultiheadAttention(
          (out_proj): NonDynamicallyQuantizableLinear(in_features=50, out_features=50, bias=True)
        )
        (linear1): Linear(in_features=50, out_features=128, bias=True)
        (dropout): Dropout(p=0.1, inplace=False)
        (linear2): Linear(in_features=128, out_features=50, bias=True)
        (norm1): LayerNorm((50,), eps=1e-05, elementwise_affine=True)
        (norm2): LayerNorm((50,), eps=1e-05, elementwise_affine=True)
        (dropout1): Dropout(p=0.1, inplace=False)
        (dropout2): Dropout(p=0.1, inplace=False)
      )
    )
  )
  (classifier): Sequential(
    (0): Linear(in_features=50, out_features=25, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.1, inplace=False)
    (3): Linear(in_features=25, out_features=5, bias=True)
  )
)
```

Classification report

Classification Report:				
	precision	recall	f1-score	support
very negative	0.5172	0.1075	0.1780	279
negative	0.4739	0.5877	0.5247	633
neutral	0.3242	0.1825	0.2336	389
positive	0.3600	0.6529	0.4641	510
very positive	0.6054	0.3383	0.4341	399
accuracy			0.4258	2210
macro avg	0.4561	0.3738	0.3669	2210
weighted avg	0.4505	0.4258	0.3993	2210

My hyperparameters using glove.6B.300d

- Num_encoder_layers = 3
- Embedding dimension 300
- nhead=4
- Optimizer: Adam
- num_epochs=50
- batch_size=48
- Droup_out = 0.1
- Learning_rate = 0.001

Model architecture

```
TransformerSentimentClassifier(  
    (embedding): Embedding(16581, 300)  
    (pos_encoder): PositionalEncoding(  
        (dropout): Dropout(p=0.1, inplace=False)  
    )  
    (transformer_encoder): TransformerEncoder(  
        (layers): ModuleList(  
            (0-2): 3 x TransformerEncoderLayer(  
                (self_attn): MultiheadAttention(  
                    (out_proj): NonDynamicallyQuantizableLinear(in_features=300, out_features=300, bias=True)  
                )  
                (linear1): Linear(in_features=300, out_features=128, bias=True)  
                (dropout): Dropout(p=0.1, inplace=False)  
                (linear2): Linear(in_features=128, out_features=300, bias=True)  
                (norm1): LayerNorm((300,), eps=1e-05, elementwise_affine=True)  
                (norm2): LayerNorm((300,), eps=1e-05, elementwise_affine=True)  
                (dropout1): Dropout(p=0.1, inplace=False)  
                (dropout2): Dropout(p=0.1, inplace=False)  
            )  
        )  
    )  
    (classifier): Sequential(  
        (0): Linear(in_features=300, out_features=150, bias=True)  
        (1): ReLU()  
        (2): Dropout(p=0.1, inplace=False)  
        (3): Linear(in_features=150, out_features=5, bias=True)  
    )  
)
```

Classification report

	precision	recall	f1-score	support
very negative	0.0000	0.0000	0.0000	279
negative	0.0000	0.0000	0.0000	633
neutral	0.0000	0.0000	0.0000	389
positive	0.2308	1.0000	0.3750	510
very positive	0.0000	0.0000	0.0000	399
accuracy			0.2308	2210
macro avg	0.0462	0.2000	0.0750	2210
weighted avg	0.0533	0.2308	0.0865	2210

Confusion Matrix:
Labels order: very negative, negative, neutral, positive, very positive
Rows = True labels, Columns = Predicted labels

```
[
  [ 0  0  0 279  0]
  [ 0  0  0 633  0]
  [ 0  0  0 389  0]
  [ 0  0  0 510  0]
  [ 0  0  0 399  0]]
```

Section 2.2.1 - BERT Fine tuning on SST-5

I edited the sequence_classification Jupyter notebook file provided to work for the SST-5 dataset. And I ran it on Colab (Upgraded to Pro subscription for students)

Here are my training hyperparameters:

```
Training Arguments:
Learning Rate: 2e-05
Batch Size: 16
Epochs: 3
Weight Decay: 0.01
Warmup Steps: 500
Trainer initialized successfully!
Starting training...
```

Here is the loss after each epoch

Epoch	Training Loss	Validation Loss	Accuracy
1	1.178400	1.125291	0.512217
2	1.027300	1.108752	0.505882
3	0.700100	1.198388	0.523529

Training completed!

Here are the results on accuracy and classification

Final Test Accuracy: 0.5235

Detailed Classification Report:

	precision	recall	f1-score	support
very negative	0.5415	0.4444	0.4882	279
negative	0.5781	0.5671	0.5726	633
neutral	0.3510	0.3573	0.3541	389
positive	0.4967	0.5941	0.5411	510
very positive	0.6554	0.5815	0.6162	399
accuracy			0.5235	2210
macro avg	0.5245	0.5089	0.5144	2210
weighted avg	0.5287	0.5235	0.5241	2210

Confusion Matrix:

Labels: [very negative, negative, neutral, positive, very positive]

```
[[124 119 27 9 0]
 [ 93 359 135 44 2]
 [ 12 127 139 104 7]
 [ 0 13 81 303 113]
 [ 0 3 14 150 232]]
```

Confusion Matrix (with labels):

	very negative	negative	neutral	positive	very positive
very negative	124	119	27	9	0
negative	93	359	135	44	2
neutral	12	127	139	104	7
positive	0	13	81	303	113
very positive	0	3	14	150	232

Task 2.2.3.2

Running the classification task on SST-5 dataset with Gemma-3-1b

=====

PERFORMANCE METRICS

=====

Classification Report:

	precision	recall	f1-score	support
very negative	0.00	0.00	0.00	279
negative	0.26	0.19	0.22	633
neutral	0.18	0.79	0.29	389
positive	0.00	0.00	0.00	510
very positive	0.14	0.00	0.00	399
accuracy			0.20	2210
macro avg	0.12	0.20	0.10	2210
weighted avg	0.13	0.20	0.12	2210

Accuracy Score: 0.1959

Task 2.2.3.3 Sentiment Classifier with Explanation

Classification Report:

	precision	recall	f1-score	support
very negative	0.20	0.10	0.14	276
negative	0.49	0.59	0.53	628
neutral	0.27	0.17	0.21	388
positive	0.38	0.60	0.47	508
very positive	0.34	0.21	0.26	397
accuracy				0.39 2197
macro avg	0.34	0.34	0.32	2197
weighted avg	0.36	0.39	0.36	2197

Accuracy Score:

0.3891670459717797

Confusion Matrix:

Labels order: very negative, negative, neutral, positive, very positive

Rows = True labels, Columns = Predicted labels

```
[[ 28 186  28  19  15]
 [ 56 368  97  78  29]
 [ 27 131  67 132  31]
 [ 17  49  47 307  88]
 [  9  20  11 272  85]]
```