Assignment-3 Report
GROUP 20

Task1

Preprocessing steps:-

Line Tokenization:

- Each line is split using .split() (space-based word tokenization).
- Tokenized data used to create training examples.

Vocabulary Construction:

- Vocabulary built from all unique tokens in the training data.
- Special tokens added: <PAD>, <START>, <STOP>, <UNK>

Token-to-ID Mapping:

- Constructed tokenizer (token → ID) and tokenizer_inv (ID → token).
- Tokens not found in vocabulary are mapped to <UNK>.

Input/Output Preparation

- For each sentence, created:
    a. input = tokens[:-1]
    b. output = tokens[1:] (shifted one step right)
- These are used for next-token prediction.

Padding and Masking:

- Sequences padded to uniform length using <PAD> token.
- Used causal masking to ensure model sees only previous tokens during training.

Created (input,output) pairs and they are wrapped using a Pytorch Dataset class for every batching

Transformer Model-architecture

Component:-

Embedding layer:- Converts tokens to dense vectors

Positional Embedding:- Learnable, gives word order awareness

Transformer Blocks:- each has Multihead Attention, feed-forward Network, Residual Connections, Layer Normalization

Output Linear Layer:- Converts hidden states to logits over vocabulary

2.2 Hyperparameters Used

Hyperparameter	Value
embed_dim	128
num_heads	4
ff_dim	256
num_layers	6
dropout	0.2
max_len	128
batch_size	16
epochs	10
lr	1e-4
label_smoothing	0.1

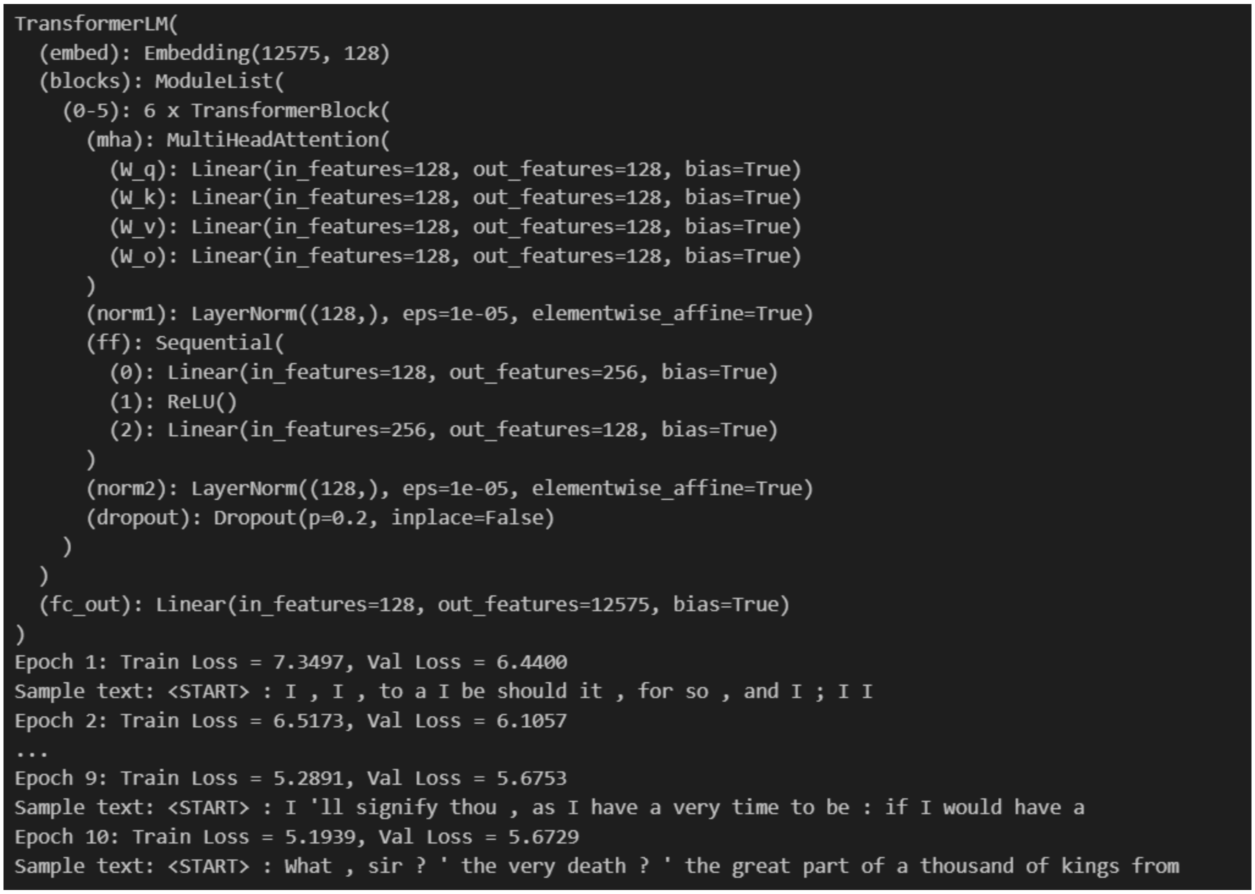
Optimizer = Adam
Casual Masking: Used in attention to prevent information leakage
Passing mask: <PAD> tokens are ignored in loss and perplexity

Training and evaluation

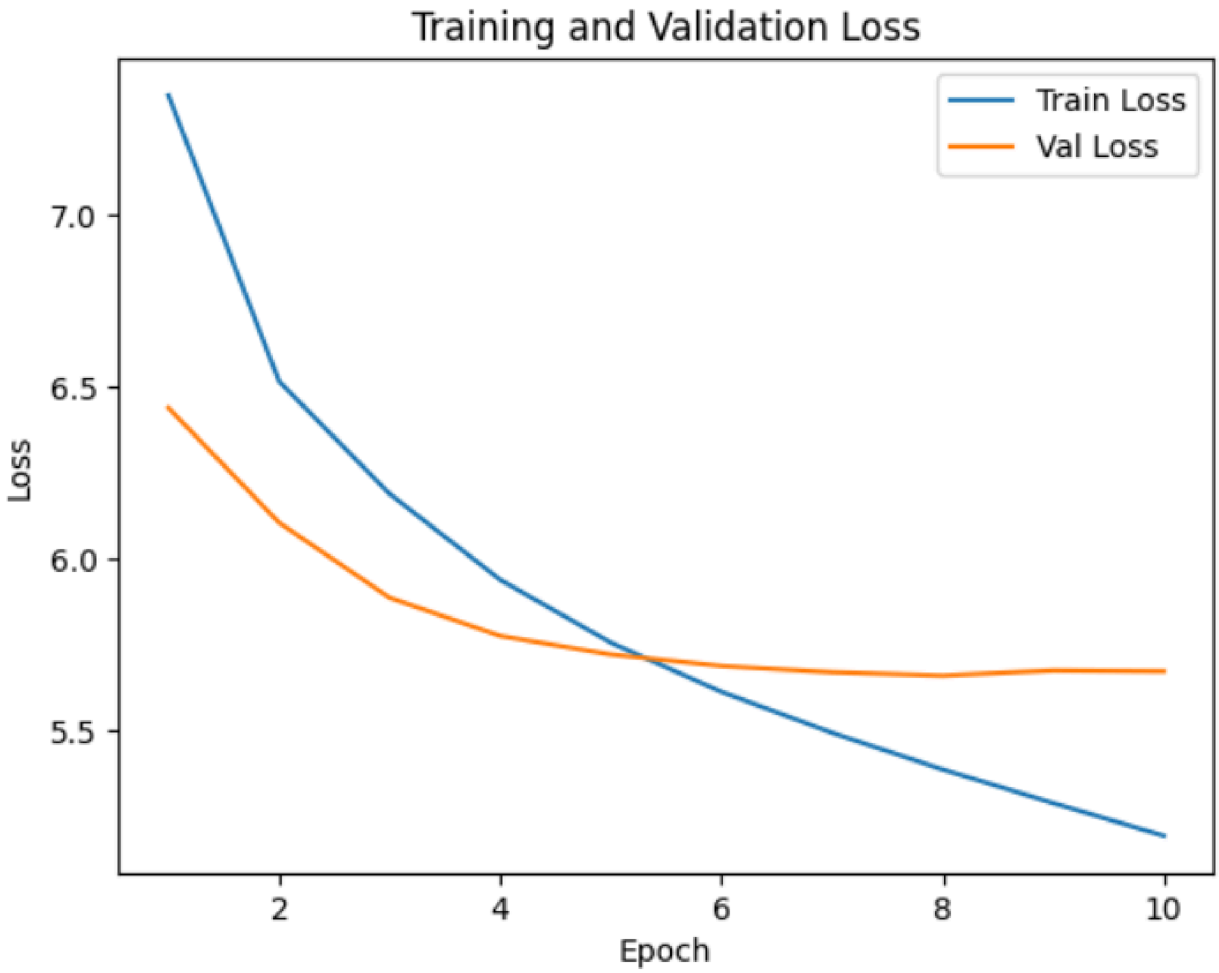
- Train the model for 10 epochs
- Log and store training and validation loss
- Generate a sample output after each epoch

Used crossEntropyLoss with label smoothing=0.1 and ignore_index=tokenizer[“<PAD>”] to avoid padding bias

Plot



```
TransformerLM(
  (embed): Embedding(12575, 128)
  (blocks): ModuleList(
    (0-5): 6 x TransformerBlock(
      (mha): MultiHeadAttention(
        (W_q): Linear(in_features=128, out_features=128, bias=True)
        (W_k): Linear(in_features=128, out_features=128, bias=True)
        (W_v): Linear(in_features=128, out_features=128, bias=True)
        (W_o): Linear(in_features=128, out_features=128, bias=True)
      )
      (norm1): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (ff): Sequential(
        (0): Linear(in_features=128, out_features=256, bias=True)
        (1): ReLU()
        (2): Linear(in_features=256, out_features=128, bias=True)
      )
      (norm2): LayerNorm((128,), eps=1e-05, elementwise_affine=True)
      (dropout): Dropout(p=0.2, inplace=False)
    )
  )
  (fc_out): Linear(in_features=128, out_features=12575, bias=True)
)
Epoch 1: Train Loss = 7.3497, Val Loss = 6.4400
Sample text: <START> : I , I , to a I be should it , for so , and I ; I I
Epoch 2: Train Loss = 6.5173, Val Loss = 6.1057
...
Epoch 9: Train Loss = 5.2891, Val Loss = 5.6753
Sample text: <START> : I 'll signify thou , as I have a very time to be : if I would have a
Epoch 10: Train Loss = 5.1939, Val Loss = 5.6729
Sample text: <START> : What , sir ? ' the very death ? ' the great part of a thousand of kings from
```



Final Perplexity:

Perplexity = exp(total_loss/Non-PAD tokens)

TASK 2

Preprocessing steps.

The CLAN dataset was first preprocessed, the social media posts in the dataset was normalized and clean.

1. Expand the contractions and Abbreviations: Function contractions was used to expand english contractions and abbreviations was hard coded.
2. Cleaning the Text:

- Lowercase all inputs to maintain consistency and reduce vocabulary size.
- Remove URLs using regex.
- Stripped special characters like emojis or punctuations.
- Reduced extra white space to avoid irregular input formats.

Model architecture and hyperparameters used.

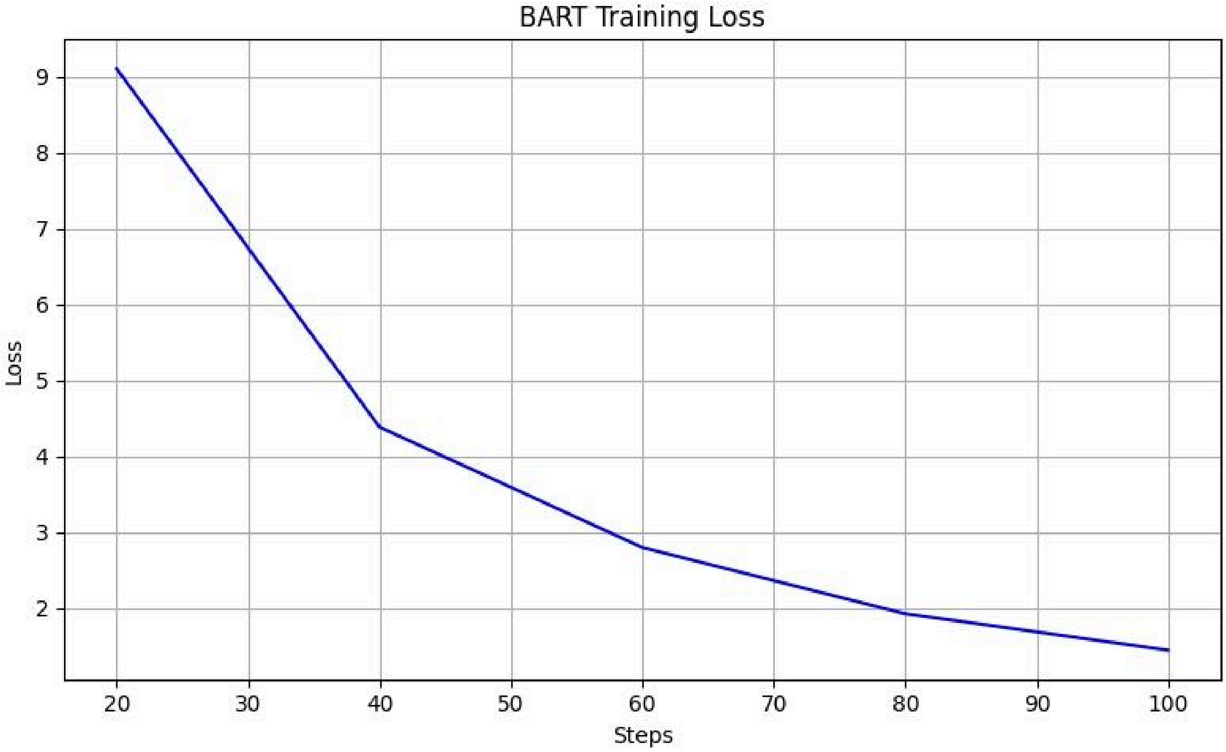
Two transformer based sequence to sequence models were trained:

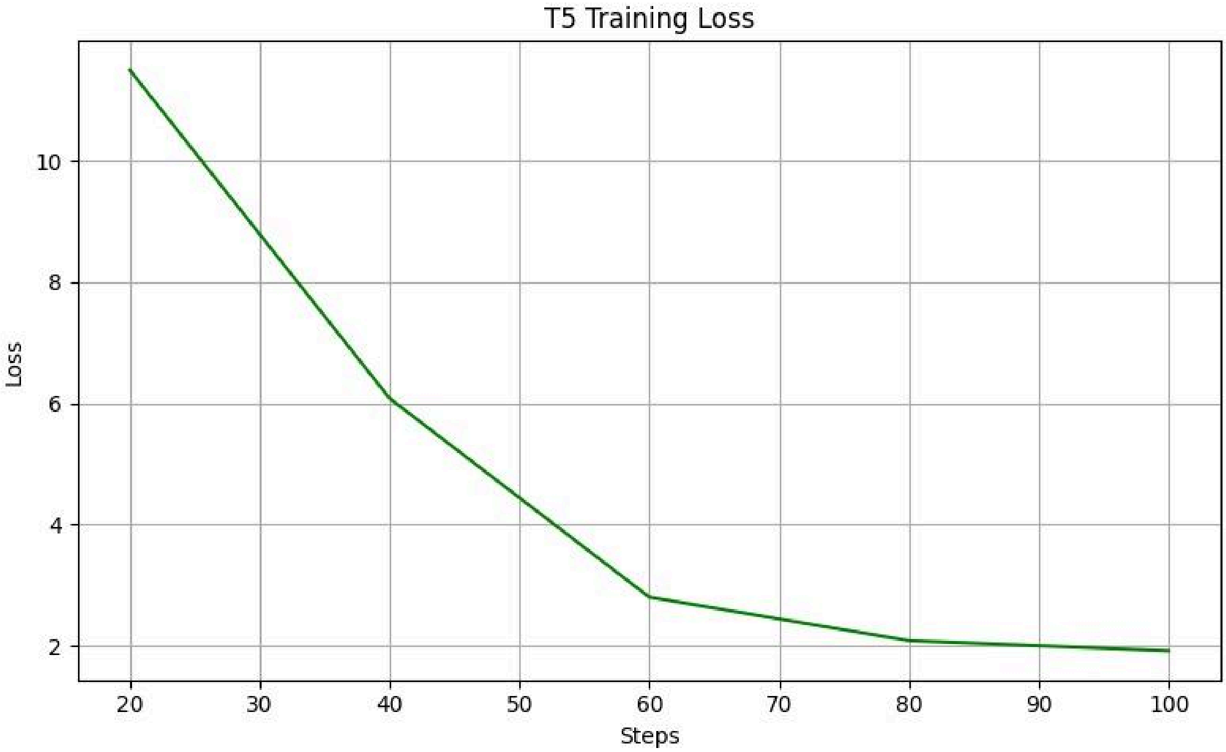
1. BART (facebook/bart-base) : Encoder Decoder architecture, Handles longer context better due to bidirectional encoding.
2. T5 (t5-small) : Text to Text transformer model, Converts all tasks into a text to text format.

Common Hyperparameters:

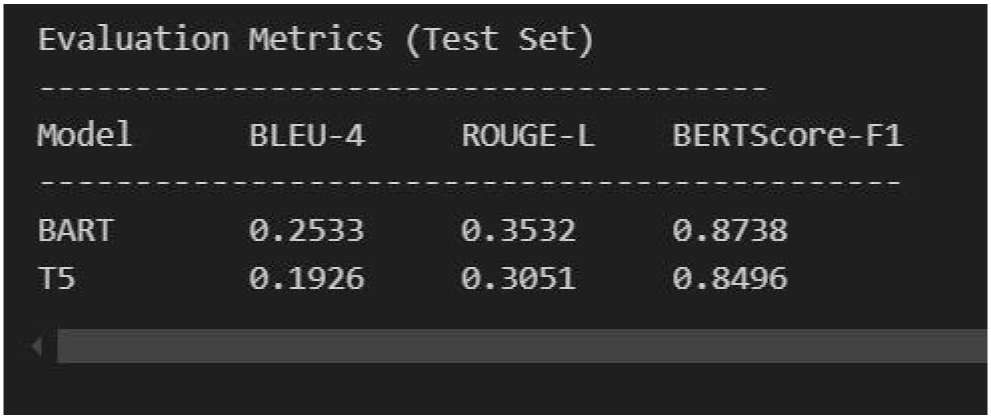
- Learning Rate: 5e-5
- Batch Size: 8
- Epochs: 3
- Weight Decay: 0.01
- Save steps: 500
- Logging steps: 50

Training loss plots for the models.





Evaluation metrics on the test set for both models. (CALCULATED FOR 5 EPOCHS)



```
Evaluation Metrics (Test Set)
----------------------------------------
Model       BLEU-4     ROUGE-L    BERTScore-F1
-----------------------------------------------
BART        0.2533     0.3532     0.8738
T5          0.1926     0.3051     0.8496
```

A comparative analysis of the performance of both models.

BART consistently outperformed T5 across all three evaluation metrics:

1. **BLEU-4:** BART achieved a higher n-gram overlap (0.2533) compared to T5 (0.1926), suggesting better fluency and token-level alignment with the reference claims.

2. **ROUGE-L:** BART's score of 0.3532 indicates it captures more relevant phrases and longer overlapping sequences than T5.

3. **BERTScore-F1:** BART reached 0.8738, reflecting better semantic similarity with the reference outputs compared to T5’s 0.8496.

Discussion on resource constraints and how they influenced model selection.

- **Model Size:** The base variants of BART and T5 were used to ensure training completed within the time and memory constraints of free-tier hardware.
- **Batch Size & Sequence Length:** Smaller batch sizes (e.g., 8) and limited maximum token lengths (128–256) were used to fit models in memory during training and inference.
- **Training Time:** Models were trained for fewer epochs with early stopping based on validation loss to prevent overfitting and minimize runtime.

Google drive link for the whole task 2 :(https://drive.google.com/file/d/1K1grrOIAA7Vu0MvjPiZqd5BVuojCFpuX/view?usp=drive_link)

Task-3

Preprocessing steps

Initially the dataset given to us contained these following columns

- text: Sarcastic post
- explanation: Human-written explanation of sarcasm
- target_of_sarcasm: Specific focus of sarcasm
- pid: Post ID used to match image

Now this data was preprocessed and changed according to the turbo model requirement like this:

Text preprocessing:

Instead of feeding only the original text, we created an **enriched input sequence** that combines:

- the sarcastic post
- its visual description
- and the target of sarcasm

This format mimics **Section 4.3** of the MuSE paper where multiple input modalities (textual and visual) and annotations (target) are **concatenated** into a single input sequence.

Image preprocessing:

Each image was processed as follows:

Step	Details
File Access	Used pid to locate the image (e.g., images/{pid}.jpg).
Resizing	Transformed to 224×224 resolution.
Normalization	Normalized with mean = [0.5] and std = [0.5].
Feature Extraction (ViT)	Passed through Vision Transformer (ViT) to get [1, 768] features.

Pooling	Mean-pooled across patches to get a [768] vector.

This aligns with the requirement to **extract high-level image features** using a Vision Transformer (ViT).

In addition to visual embeddings:

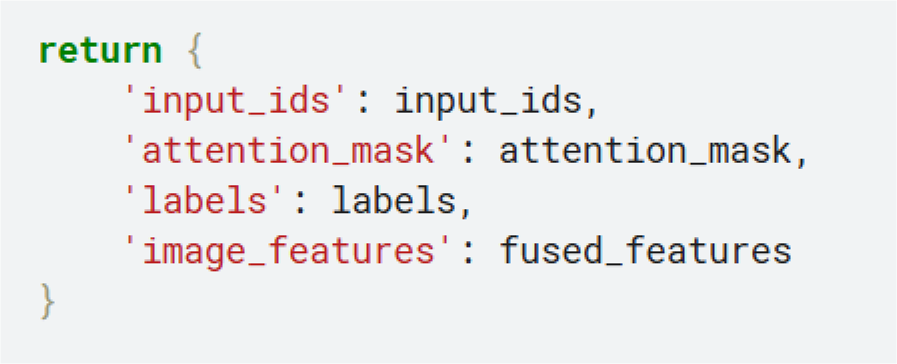
- We used D_train.pkl to load image descriptions.
- We used O_train.pkl to load detected object vectors.

The image description was **concatenated to the input text**, and detected object vectors were optionally fused or fallback embeddings.

Tokenizer Input:
We used BART's tokenizer to convert the enriched text and explanation into token IDs:
This produced:

- input_ids and attention_mask for the input sequence
- labels for the target explanation

Final Dataset Format which will give to model:



```
return {
    'input_ids': input_ids,
    'attention_mask': attention_mask,
    'labels': labels,
    'image_features': fused_features
}
```

Model Architecture and Hyperparameter used

Components:

- Text Encoder: BART encoder (facebook/bart-base)

- Image Encoder: Vision Transformer (ViT)
- Fusion Module: SharedFusion combines text & visual features
- Decoder: BART decoder for generating sarcastic explanations

Fusion:

- Extract mean-pooled text and image embeddings
- Project both to a common space using linear layers
- Concatenate and project again to get fused representation

MuSE class:

- In which we are preparing the data to the model input as explained above.

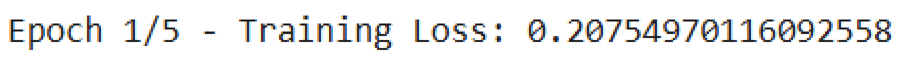
TurboModel Class

- This wraps both BART and the fusion mechanism.

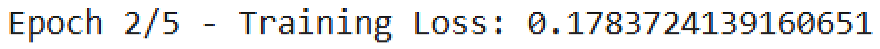
HyperParameters used

Hyperparameter	Value
BART Model	facebook/bart-base
Vit Model	vit-base-patch16-224-in21k
Image Size	224 x 224
Token Max Length	128
Fusion Dim	768
Optimizer	AdamW
LR	5e-5
Batch Size	16
Activation Fun	ReLU

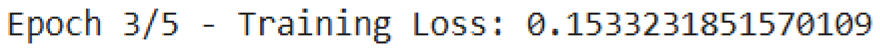
Training loss after each epoch:



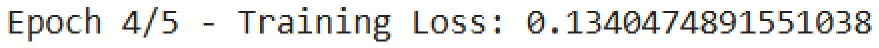
```
Epoch 1/5 - Training Loss: 0.20754970116092558
```



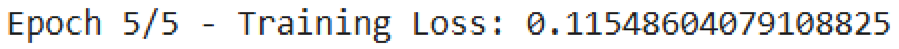
```
Epoch 2/5 - Training Loss: 0.1783724139160651
```



```
Epoch 3/5 - Training Loss: 0.1533231851570109
```

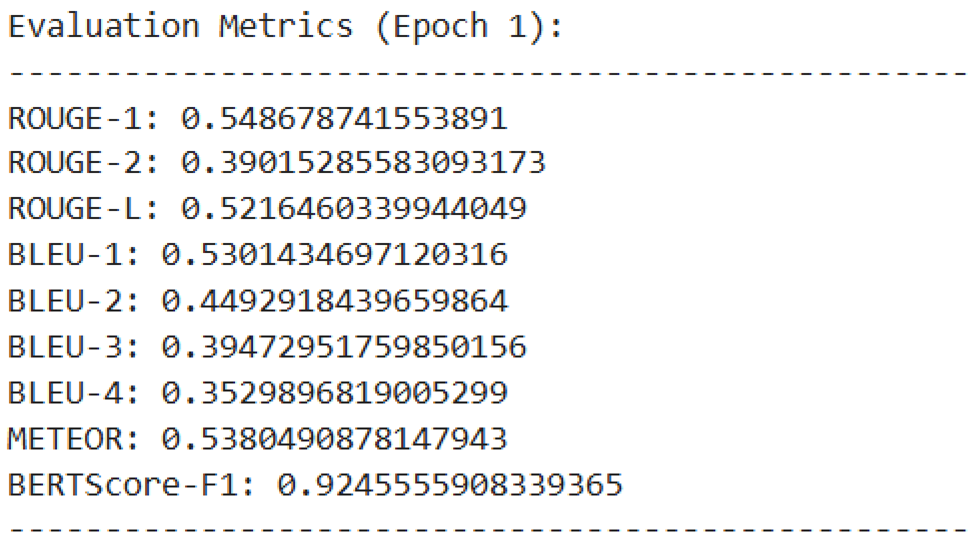


```
Epoch 4/5 - Training Loss: 0.1340474891551038
```

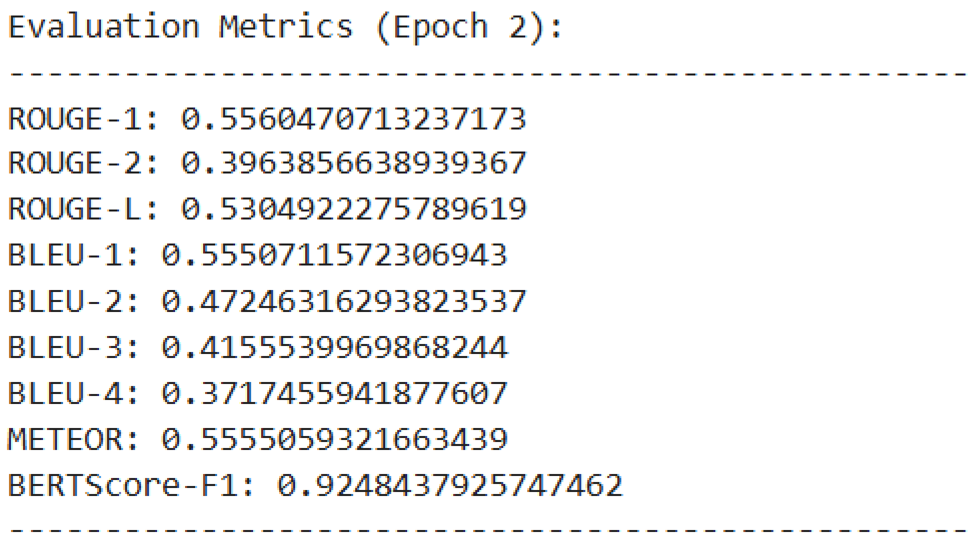


```
Epoch 5/5 - Training Loss: 0.11548604079108825
```

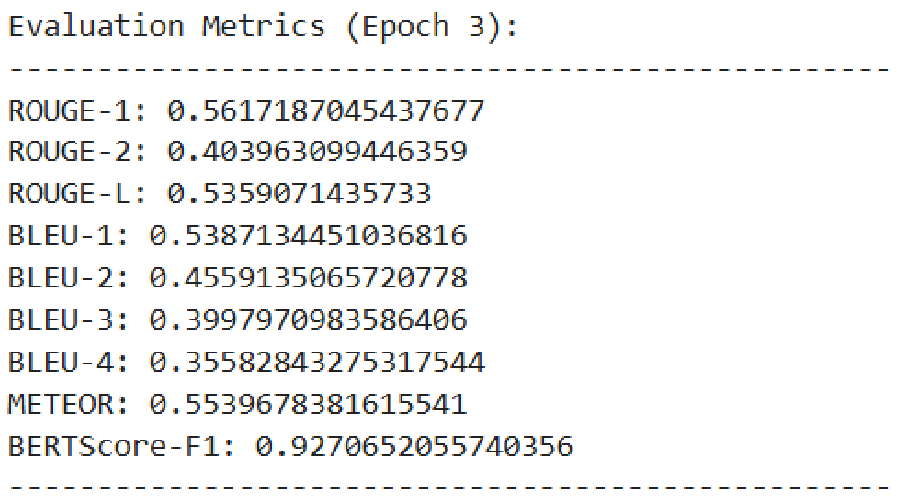
Evaluation metrics on the validation set after each epoch:



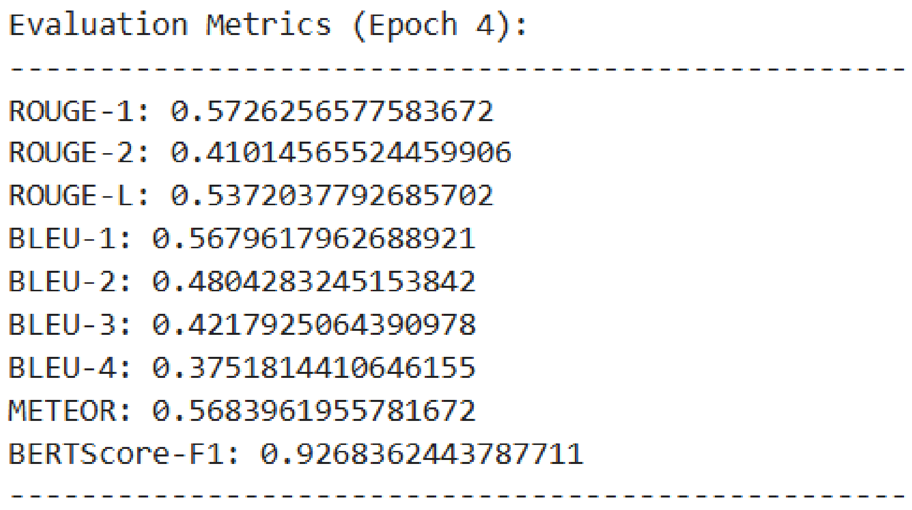
```
Evaluation Metrics (Epoch 1):
--------------------------------------------------
ROUGE-1: 0.548678741553891
ROUGE-2: 0.39015285583093173
ROUGE-L: 0.5216460339944049
BLEU-1: 0.5301434697120316
BLEU-2: 0.4492918439659864
BLEU-3: 0.39472951759850156
BLEU-4: 0.3529896819005299
METEOR: 0.5380490878147943
BERTScore-F1: 0.9245555908339365
--------------------------------------------------
```



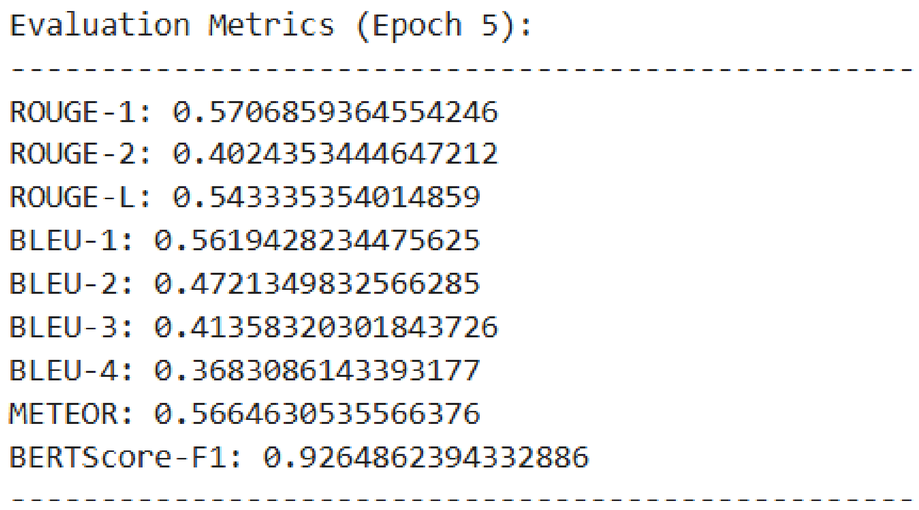
```
Evaluation Metrics (Epoch 2):
--------------------------------------------------
ROUGE-1: 0.5560470713237173
ROUGE-2: 0.3963856638939367
ROUGE-L: 0.5304922275789619
BLEU-1: 0.5550711572306943
BLEU-2: 0.47246316293823537
BLEU-3: 0.4155539969868244
BLEU-4: 0.3717455941877607
METEOR: 0.5555059321663439
BERTScore-F1: 0.9248437925747462
--------------------------------------------------
```



```
Evaluation Metrics (Epoch 3):
--------------------------------------------------
ROUGE-1: 0.5617187045437677
ROUGE-2: 0.403963099446359
ROUGE-L: 0.5359071435733
BLEU-1: 0.5387134451036816
BLEU-2: 0.4559135065720778
BLEU-3: 0.3997970983586406
BLEU-4: 0.35582843275317544
METEOR: 0.5539678381615541
BERTScore-F1: 0.9270652055740356
--------------------------------------------------
```



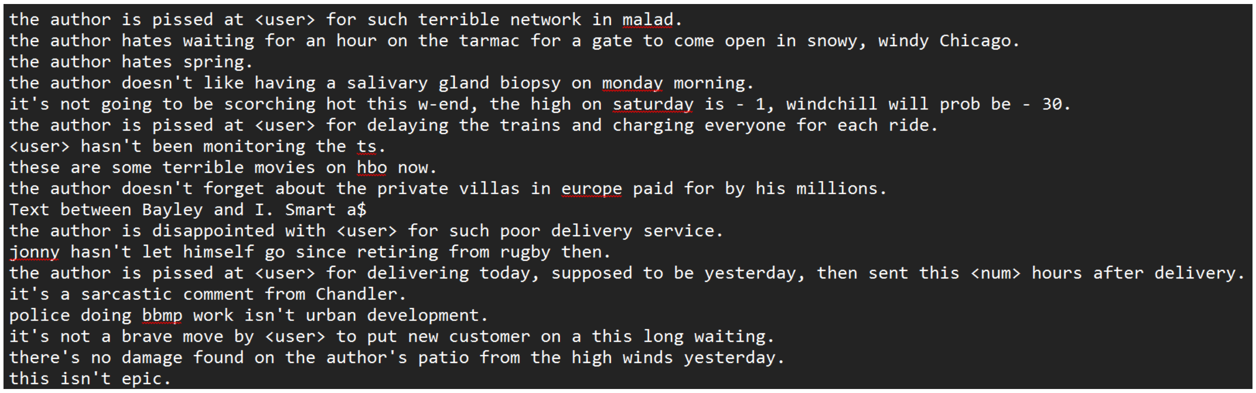
```
Evaluation Metrics (Epoch 4):
--------------------------------------------------
ROUGE-1: 0.5726256577583672
ROUGE-2: 0.41014565524459906
ROUGE-L: 0.5372037792685702
BLEU-1: 0.5679617962688921
BLEU-2: 0.4804283245153842
BLEU-3: 0.4217925064390978
BLEU-4: 0.3751814410646155
METEOR: 0.5683961955781672
BERTScore-F1: 0.9268362443787711
--------------------------------------------------
```



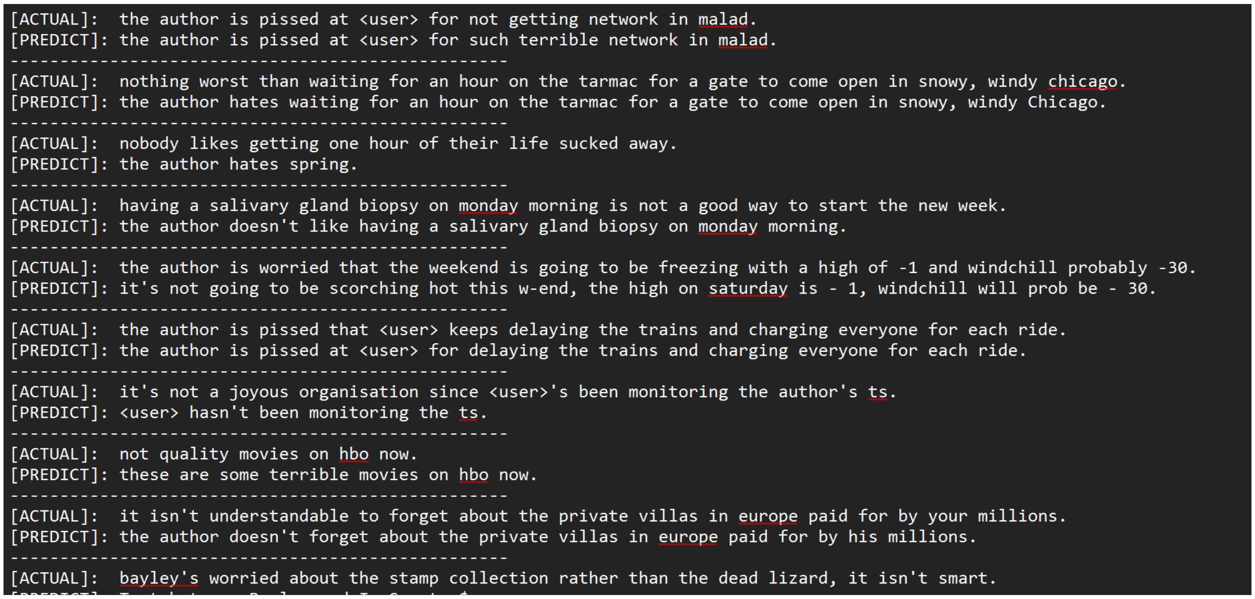
```
Evaluation Metrics (Epoch 5):
--------------------------------------------------
ROUGE-1: 0.5706859364554246
ROUGE-2: 0.4024353444647212
ROUGE-L: 0.543335354014859
BLEU-1: 0.5619428234475625
BLEU-2: 0.4721349832566285
BLEU-3: 0.41358320301843726
BLEU-4: 0.3683086143393177
METEOR: 0.5664630535566376
BERTScore-F1: 0.9264862394332886
--------------------------------------------------
```

We choose the max epoch to be 5 because after this epoch the scores were not increasing so there is no point to train beyondepoch 5.

These were the explanations we got on Best model (downloaded in txt file)



```
the author is pissed at <user> for such terrible network in malad.
the author hates waiting for an hour on the tarmac for a gate to come open in snowy, windy Chicago.
the author hates spring.
the author doesn't like having a salivary gland biopsy on monday morning.
it's not going to be scorching hot this w-end, the high on saturday is - 1, windchill will prob be - 30.
the author is pissed at <user> for delaying the trains and charging everyone for each ride.
<user> hasn't been monitoring the ts.
these are some terrible movies on hbo now.
the author doesn't forget about the private villas in europe paid for by his millions.
Text between Bayley and I. Smart a$
the author is disappointed with <user> for such poor delivery service.
jonny hasn't let himself go since retiring from rugby then.
the author is pissed at <user> for delivering today, supposed to be yesterday, then sent this <num> hours after delivery.
it's a sarcastic comment from Chandler.
police doing bbmp work isn't urban development.
it's not a brave move by <user> to put new customer on a this long waiting.
there's no damage found on the author's patio from the high winds yesterday.
this isn't epic.
```



```
[ACTUAL]:  the author is pissed at <user> for not getting network in malad.
[PREDICT]: the author is pissed at <user> for such terrible network in malad.
--------------------------------------------------
[ACTUAL]:  nothing worst than waiting for an hour on the tarmac for a gate to come open in snowy, windy chicago.
[PREDICT]: the author hates waiting for an hour on the tarmac for a gate to come open in snowy, windy Chicago.
--------------------------------------------------
[ACTUAL]:  nobody likes getting one hour of their life sucked away.
[PREDICT]: the author hates spring.
--------------------------------------------------
[ACTUAL]:  having a salivary gland biopsy on monday morning is not a good way to start the new week.
[PREDICT]: the author doesn't like having a salivary gland biopsy on monday morning.
--------------------------------------------------
[ACTUAL]:  the author is worried that the weekend is going to be freezing with a high of -1 and windchill probably -30.
[PREDICT]: it's not going to be scorching hot this w-end, the high on saturday is - 1, windchill will prob be - 30.
--------------------------------------------------
[ACTUAL]:  the author is pissed that <user> keeps delaying the trains and charging everyone for each ride.
[PREDICT]: the author is pissed at <user> for delaying the trains and charging everyone for each ride.
--------------------------------------------------
[ACTUAL]:  it's not a joyous organisation since <user>'s been monitoring the author's ts.
[PREDICT]: <user> hasn't been monitoring the ts.
--------------------------------------------------
[ACTUAL]:  not quality movies on hbo now.
[PREDICT]: these are some terrible movies on hbo now.
--------------------------------------------------
[ACTUAL]:  it isn't understandable to forget about the private villas in europe paid for by your millions.
[PREDICT]: the author doesn't forget about the private villas in europe paid for by his millions.
--------------------------------------------------
[ACTUAL]:  bayley's worried about the stamp collection rather than the dead lizard, it isn't smart.
```

Explanation of section 3.3 and section 3.4:

3.3

- Extract high-level image features using a Vision Transformer (ViT)
  We employ the pretrained ViTModel from Hugging Face (google/vit-base-patch16-224-in21k) to extract high-level features from images. Each image is resized to 224×224 and normalized.

- Concatenate token sequences
  We concatenate various text components including the post text, its corresponding image description, and the sarcasm target to form an enriched input sequence. This format helps the model jointly learn from both context and visual cues for generating sarcasm explanations.

- Incorporate sarcasm target into explanation generation
  The sarcasm target is explicitly included in the input text using the `[TARGET]:` token to ensure the BART encoder is aware of the intended sarcastic target in each sample.

- Use BART base model for generation
  We used facebook/bart-base from hugging face to generate sequences.

- Implement a Shared Fusion Mechanism
  Implemented Shared Fusion Mechanism as explained above in model architecture.

3.4

- Focus on Relevant Tokens in Fusion

  “text_embeddings = self.bart.model.encoder(input_ids=input_ids, attention_mask=attention_mask).last_hidden_state.mean(dim=1)”
  When you average these, you summarize the whole sentence into a dense vector.
  This embedding retains critical context, such as:
  - Sarcasm cues
  - Sentiment flow
  - Descriptive elements

- Shared Fusion Weights Are Trainable
  As we haven't used any torch.no_grad(), hence remains trainable.

Best_Model:

https://drive.google.com/drive/folders/18p2hZVFuN5N6lwbvM5UridHhTeunAVCY

Group Members and Contributions:

Dhairya(2022157): Task-1

Harsh Vishwakarma(2022205): Task-2

Pandillapelly Harshvardhini(2022345): Task-3