CSE556: Natural Language Processing
Assignment 4

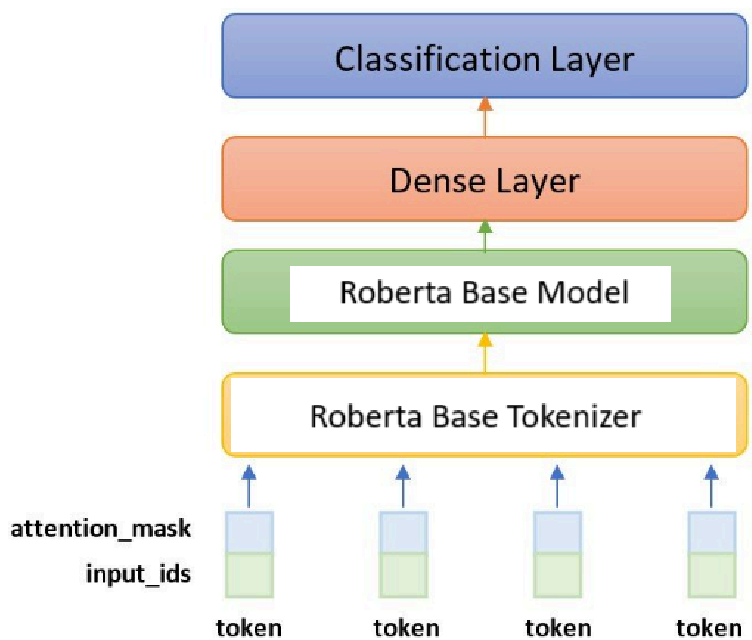
Task 1:

- **How we preprocessed the dataset:**

1. Removed the duplicate dialogues which had multiple flips and kept only 1 dialogue from such instances in both training and validation data, as the ERC task didn't depend on triggers.
2. Did unicode mapping for all the special characters.
3. Instead of passing data dialoguewise, it was passed utterance-wise, where an input instance consisted of a sequence of dialogues passed in the form as mentioned in this paper: https://arxiv.org/pdf/2108.12009.pdf

- **Loss functions and optimizer:**
    1. Optimizer: AdamW with $2 * 10^{-5}$ learning rate is used
    2. Loss function: Cross-Entropy Loss

- **Model M1:**
    1. These input sequences are passed through the "roberta-base" pretrained tokenizer, which follows a byte-pair encoding scheme.
    2. Tokenized inputs are passed through the "roberta-base" model with default parameters, followed by a linear layer on top.

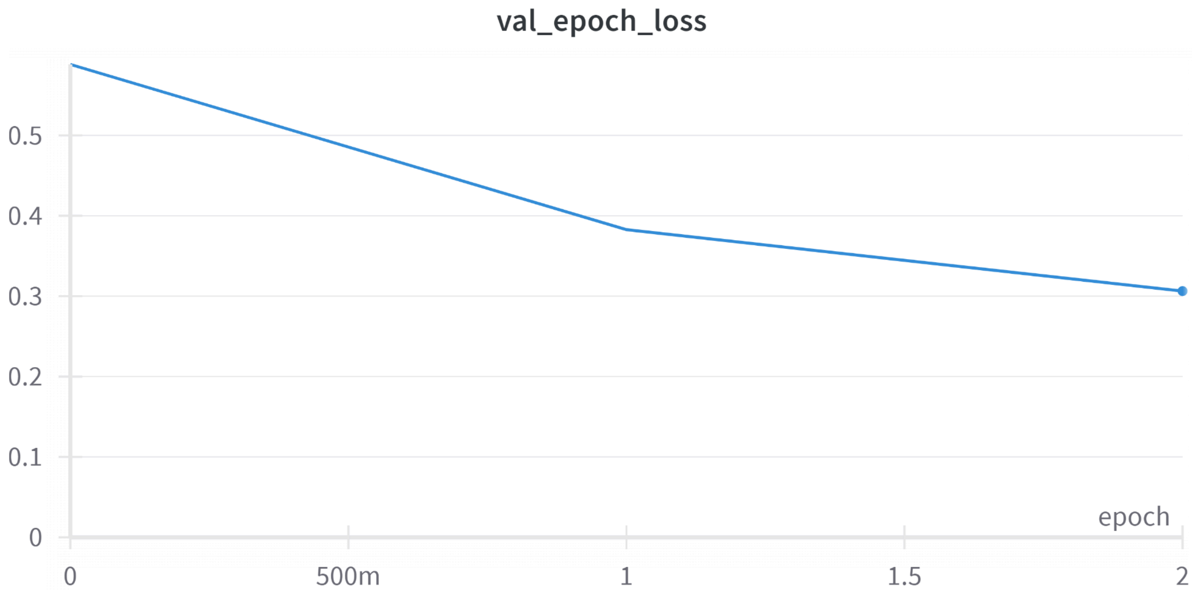


- **Model M2:**
    1. These input sequences are passed through the “roberta-base” pretrained tokenizer, which follows a byte-pair encoding scheme.
    2. Tokenized inputs are passed through the torch default embeddings with a dictionary size 50265 and 768 embedding dimension.
    3. The embeddings are passed through a 2-layer Bidirectional-GRU followed by a linear layer on top.

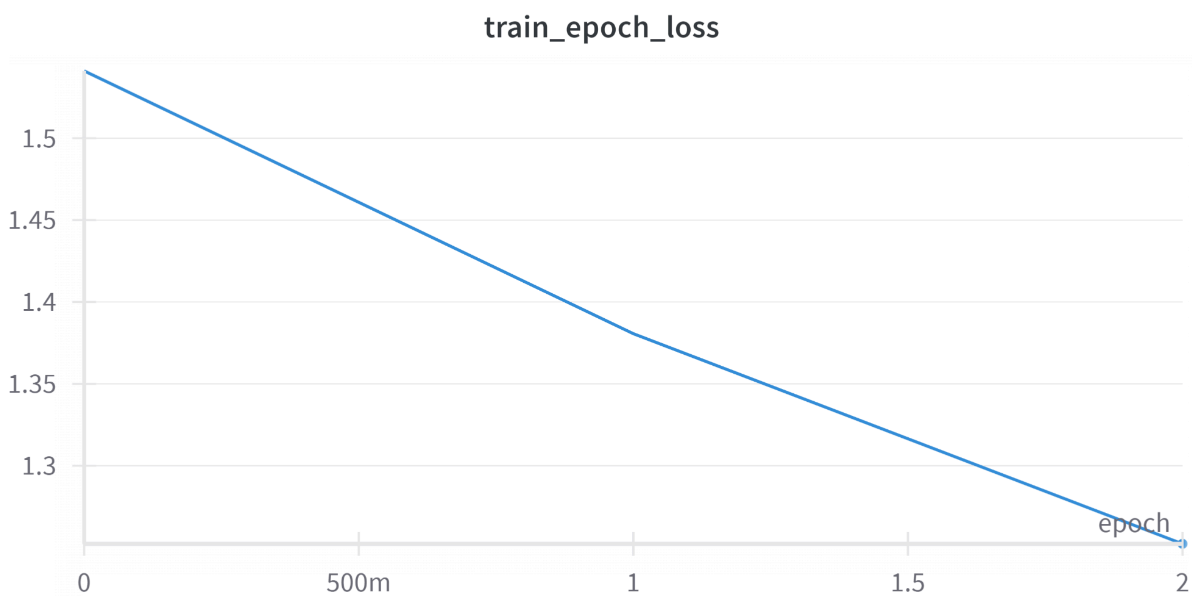
- M1 beats M2.
  RoBERTa captures contextual information, which is really important in ERC. It also encodes semantic relationships.
  GRU captures sequential patterns, but it may struggle with very long-range dependencies, as it considers only a fixed window of context.

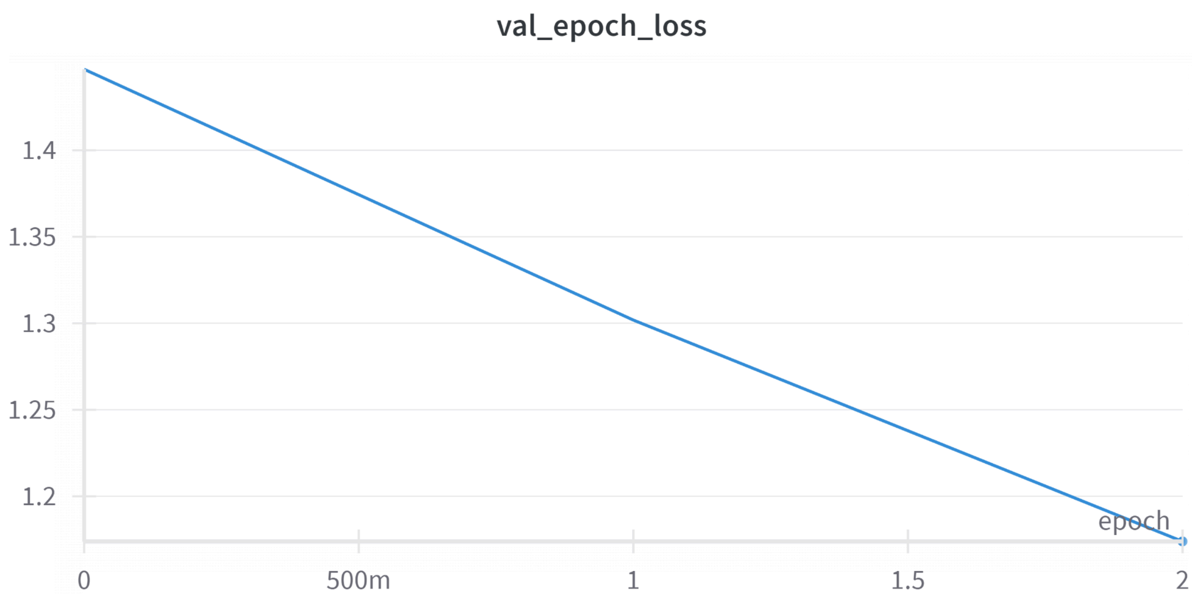
Intuition Behind Models:
For model M1, we were inspired by the EmoBERTa paper ([2108.12009.pdf (arxiv.org)](https://arxiv.org/pdf/2108.12009.pdf)). For Model M2, we wanted to see how our dataset worked with a simpler model, and taking inspiration from assignment 2, we used a bidirectional GRU.

Model M1:


val_epoch_loss
0.5
0.4
0.3
0.2
0.1
0
epoch
0
500m
1
1.5
2

Model M2:


train_epoch_loss
1.5
1.45
1.4
1.35
1.3
epoch
0
500m
1
1.5
2



Results using M1:
F1 Micro: 0.884549356223176,
F1 Macro: 0.8625106878464001,
F1 Weighted: 0.8838961470917037

Results using M2:
F1 Micro: 0.5394849785407725,
F1 Macro: 0.33780431640347536,
F1 Weighted: 0.4950421213811303

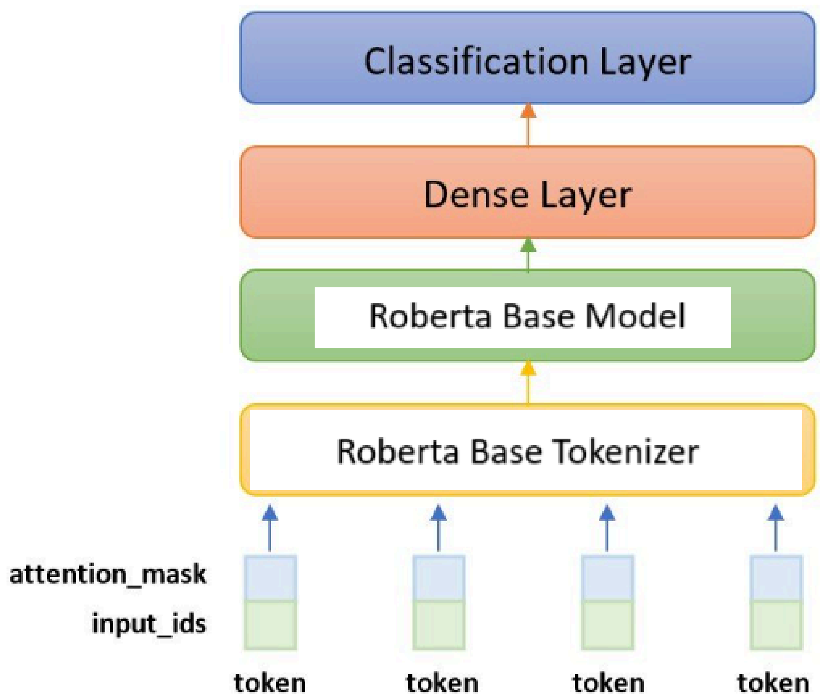
Task 2:

- **How we preprocessed the dataset:**
    1. Did Unicode mapping for all the special characters.
    2. We pass input utterance-wise with the previous context of the episode with the trigger.
       For eg.

Target -> 3rd utterance of episode “n”
We will pass three utterances with their speakers and emotions along with the trigger of 3rd utterance.

- **Loss functions and optimizer:**
  1. Optimizer: AdamW with $1 * 10^{-5}$ learning rate is used
  2. Loss function: Binary Cross-Entropy Loss

- **Model M3:**
  1. These input sequences are passed through the “roberta-base” pretrained tokenizer, which follows a byte-pair encoding scheme.
  2. Tokenized inputs are passed through the “roberta-base” model with default parameters, followed by a linear layer on top.



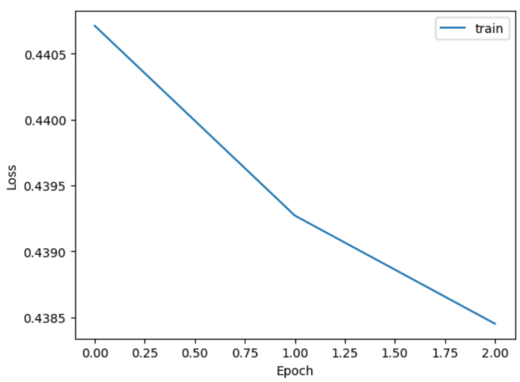
- **Model M4:**
  1. These input sequences are passed through the “roberta-base” pretrained tokenizer, which follows a byte-pair encoding scheme.
  2. Tokenized inputs are passed through the torch default embeddings with a dictionary size 50265 and 500 embedding dimension.
  3. The embeddings are passed through a 1-layer Bidirectional-GRU followed by a linear layer on top.

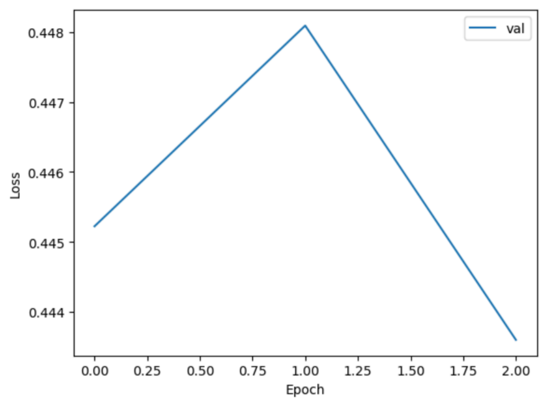
- M3 beats M4.
  RoBERTa captures contextual information, which is really important in ERC. It also encodes semantic relationships.
  GRU captures sequential patterns, but because we are RoBERTa as tokenizer and it is able to capture contextual as well as sequential patterns. Possibly due to which slight overfitting is happening in that model.

Intuition Behind Models:

For model M3, we were inspired by the EmoBERTa paper ([2108.12009.pdf (arxiv.org)]) for task 1 and because the results were good for emotion classification, we tried to use it for task 2 also. We decided to add GRU on the RoBERTa tokenizer because we can capture more sequential information using that.

Model M3:

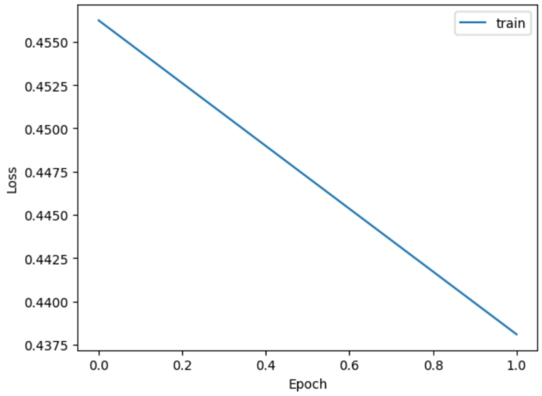


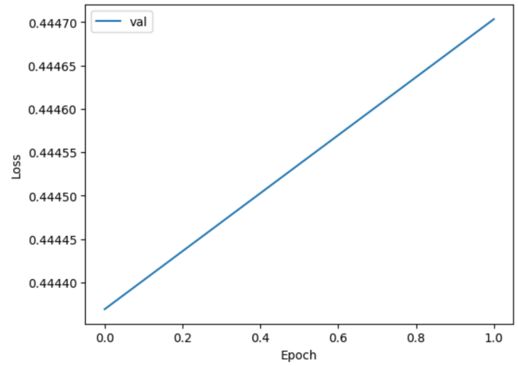


Results using Model M3

F1 micro:0.8214285714285714
F1 macro:0.45098039215686275
F1 weighted:0.7408963585434174

Model M4:





Results using Model M4

F1 micro:0.6785714285714286
F1 macro:0.4042553191489362
F1 weighted:0.5486322188449849

Contribution

Task1 : Richa and Sahil
Task2 : Vartika and Prakhar