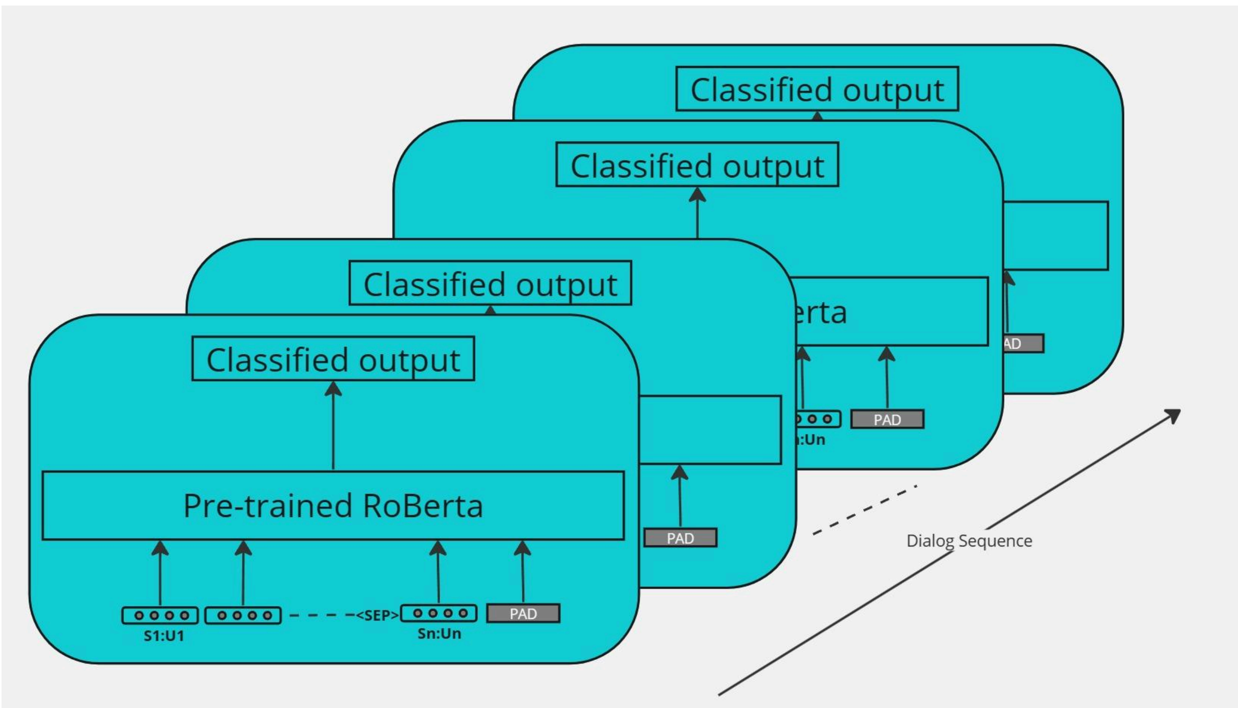
Natural Language Processing
Assignment: 4

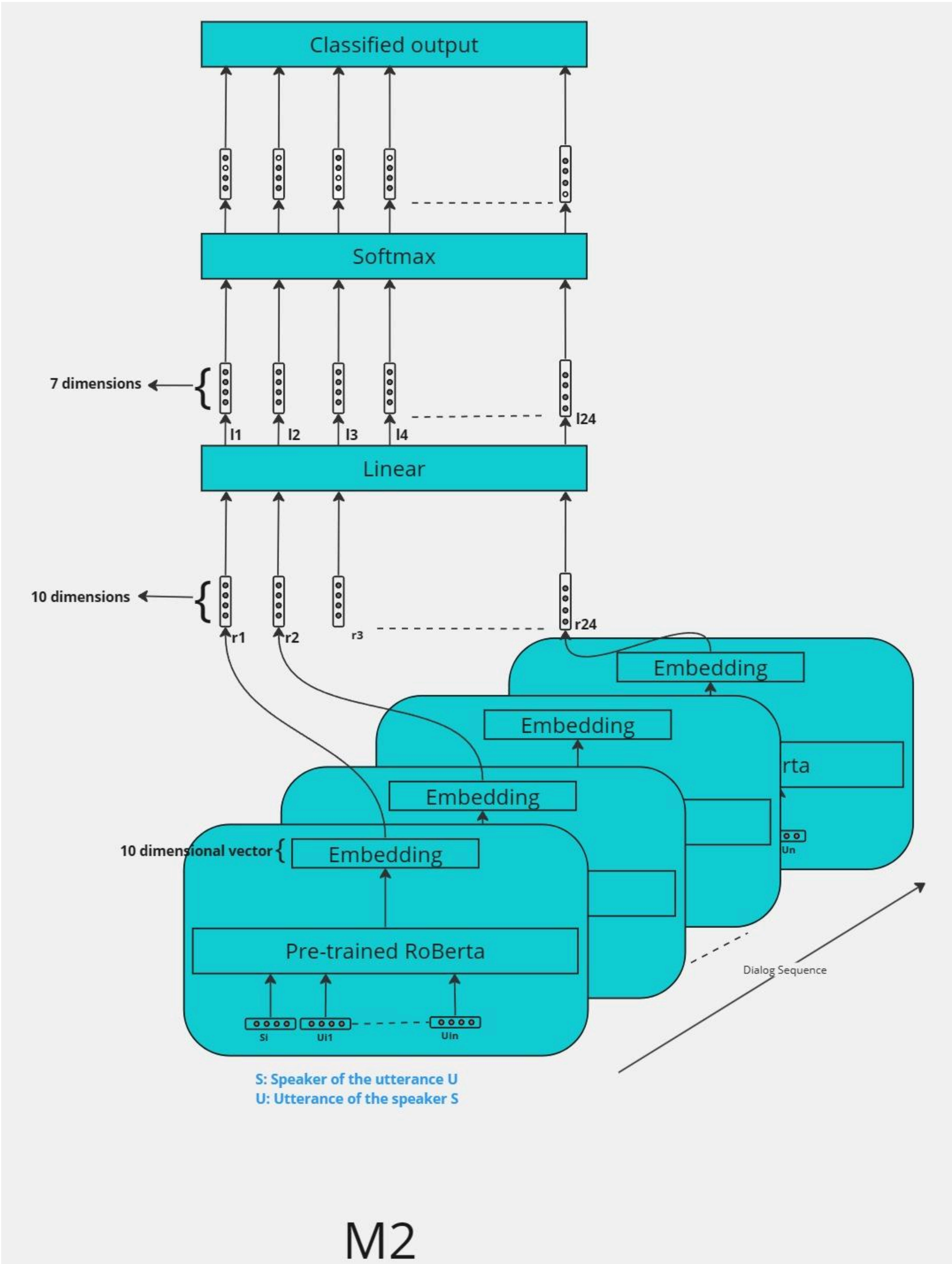
Karan Gupta (2021258) | Raghav Sakhuja (2021274) | Shivesh Gulati (2021286) | Rahul Oberoi (2021555)

TASK-1

Block Diagram M1:



Block Diagram M2:



Input: Each data instance consists of a speaker's utterance and the corresponding speaker identifier (Speaker1: Utterance1). To facilitate uniformity in input size and ensure compatibility with model requirements, utterances have been padded to a fixed length of 25 tokens. Padding is achieved using a special token recognised by the ROBERTa model “</s>”, which indicates the end of the sentence.

Intuition for M1 and M2

M1 Intuition

The intuition behind model M1: In our model we have used RoBERTa for sequence classification. The intuition was that the task was to predict the emotion of the current speaker given the previous contexts i.e. given the previous conversation. We passed the context and the current utterance separated by SEP token to the model to predict the emotion of the current utterance. To use this idea the dataset was passed in such a way that for each utterance only the previous utterances was passed as the context. The number of labels specified was 7 to predict each of the emotions. This data augmentation allowed the model to learn from the previous contexts.

Data format:

s1: utterance <sep> s2: utterance
s1: utterance s2:utterance <sep> s3: utterance
.
.
.
s1: utterance s2:utterance...sn-1:utterance <sep> sn: utterance

M2 Intuition

The intuition behind model M2: In our model architecture, we've utilised RoBERTa to extract contextual word embeddings of the input text; we've chosen to set a dropout rate of 0.07 during the training phase to avoid overfitting (the default value of dropout is 1.0 in RoBERTa, 0.07 is the value that we used and we reached this value by testing multiple dropouts). We specified the number of labels as 10, in the configuration of the Roberta Model, which resulted in embeddings of 10 dimensions, which were then passed through a linear layer and mapped to 7 logits. The final step involves applying a softmax activation function to the output of the linear layer in order to get the emotion prediction from the logits.

We use </s> to pad all the conversations to make them of equal size.

Data format:

S1:utterance

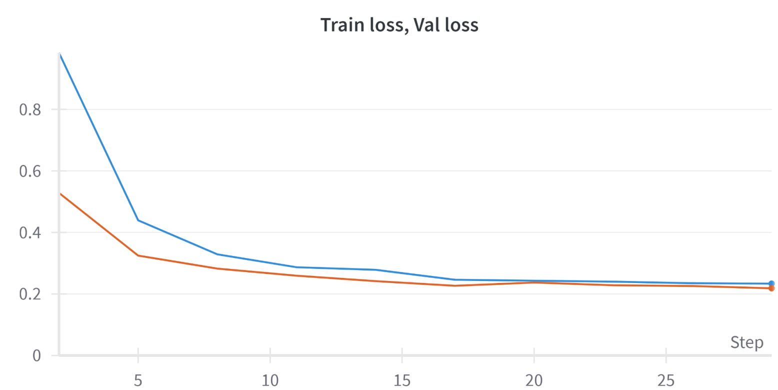
S2:utterance
S3:utterance
.
.
.
.
Sn:utterance

Better Model: M1

The model M1 is better because the model considers the previous conversation as the context for the current utterance as it is predicting the emotion of only the last utterance.The sequence classification considers the complete information of the previous conversation.

The model M2 performance is low because it is not creating contextual embeddings including the previous conversations. It is creating the embeddings for individual utterances and this doesn't capture the information of the previous utterances. M2 completely relies on the Linear layer to capture the context of the conversation.

<u>Graphs M1:</u>



Blue= train_loss
orange= val_loss

<u>Classification Report M1</u>

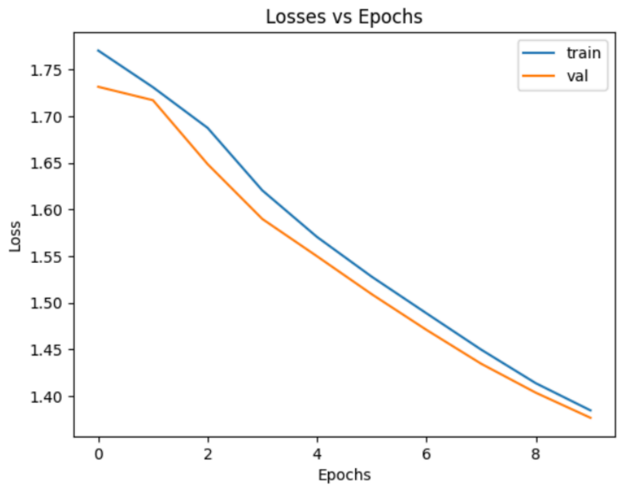
```
0.8903389945726712 0.9086820215441184
              precision    recall  f1-score   support

           0       0.94      0.87      0.91      1008
           1       0.94      0.84      0.88       558
           2       0.94      0.84      0.89       788
           3       0.87      0.84      0.86       265
           4       0.95      0.80      0.87       215
           5       0.89      0.91      0.90      1259
           6       0.90      0.96      0.93      3200

    accuracy                           0.91      7293
   macro avg       0.92      0.87      0.89      7293
weighted avg       0.91      0.91      0.91      7293
```

'surprise':0
,'sadness':1,
'Anger':2,
'Fear':3,
'Disgust':4,
'Joy':5,
'neutral':6

Graphs M2:



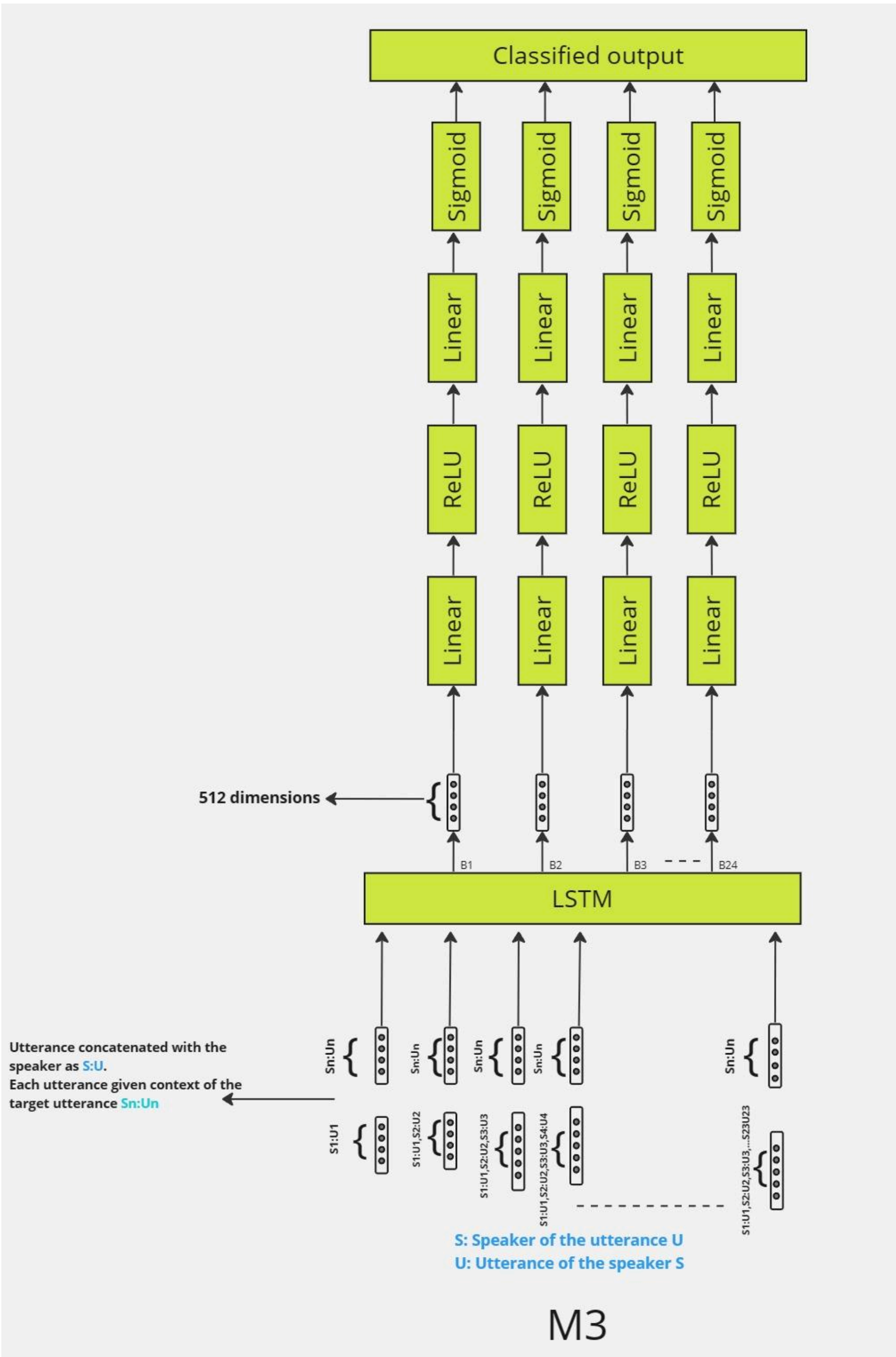
Classification Report M2:

	precision	recall	f1-score	support
0	0.83	1.00	0.91	3200
1	0.82	0.85	0.84	1008
2	0.93	0.25	0.39	558
3	0.81	0.76	0.78	788
4	1.00	0.08	0.15	265
5	0.96	0.11	0.19	215
6	0.80	0.95	0.87	1259
accuracy			0.83	7293
macro avg	0.88	0.57	0.59	7293
weighted avg	0.84	0.83	0.79	7293

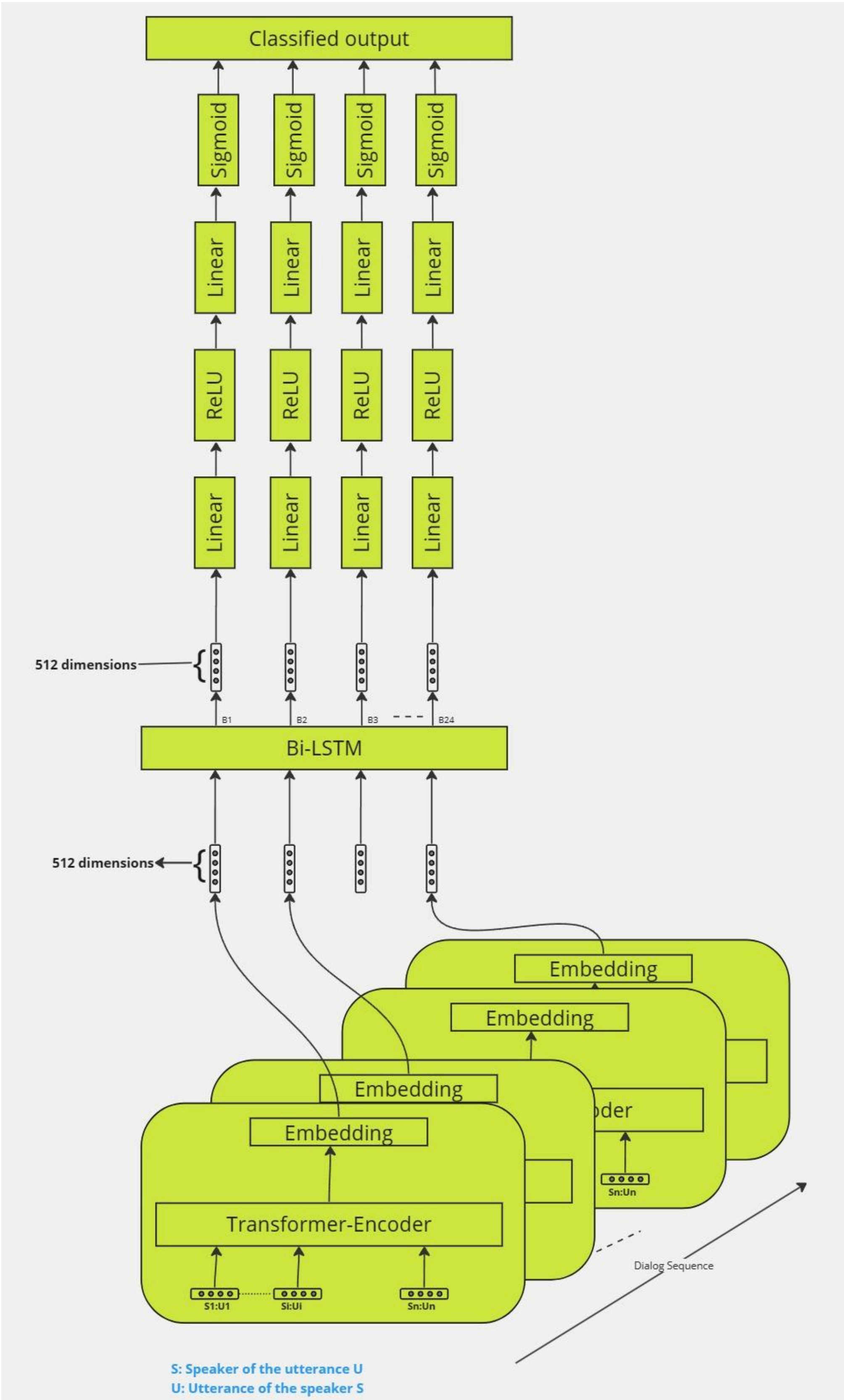
'neutral':0
'surprise':1,
'sadness':2,
'anger':3,
'fear':4,
'disgust':5,
'joy':6,

<u>TASK-2</u>

Block Diagram M3:



Block Diagram M4:



Intuition for M3 and M4

M3 Intuition

In model M3, we redefined our goal as considering each of the triggers as possible classes and converting the problem into a sequence classification problem. This was achieved by considering one entire utterance as an input to the LSTM layer at once, along with that the target sequence was also given as input. Using LSTM as our base layer allowed us to capture the flow of information through time. All conversations follow the unidirectional time aspect, wherein the conversations move forward with utterances as time progresses; hence, this was our intuition for using the LSTM layer. The output of each of the LSTM cells was then given as input to a linear layer, wherein the number of linear layers was kept equal to the LSTM cells, and the output of the linear layer was two logits so that it could be passed through a sigmoid layer to predict the class, of that utterance as either : 0-> Not a trigger 1->Trigger

Data format:

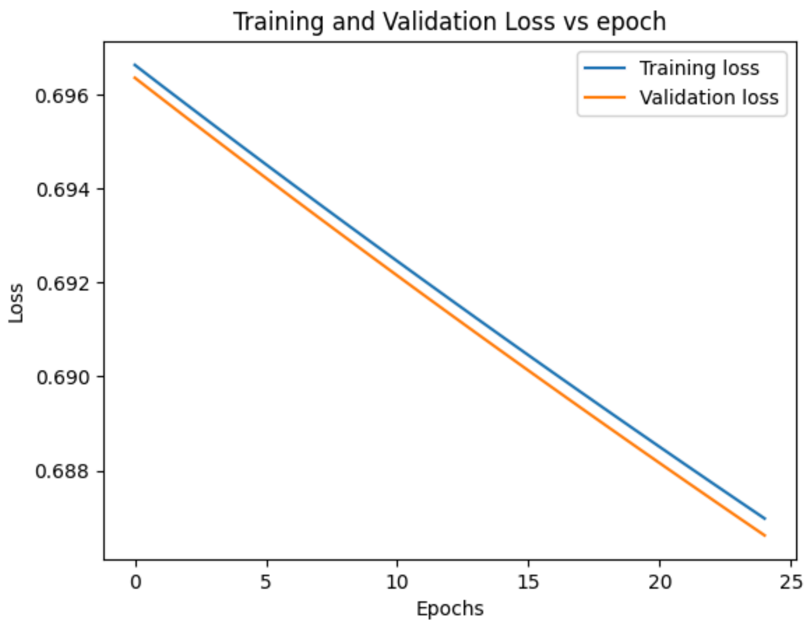
s1: utterance,	sn: utterance
s1:utterance,s2:utterance,	sn: utterance
s1:utterance,s2:utterance,s3: utterance,	sn: utterance
.	
.	
.	
s1: utterance s2:utterance...sn-1:utterance,	sn: utterance

M4 Intuition

The intuition behind model M4, is that we observed after getting the output from the LSTM, there were still a considerable number of misclassifications. Hence, we decided to add a transformer encoder layer below the LSTM with 8 attention heads and 3 encoder layers in order to get the encoded input with self-attention, leading to better context being fed into the LSTM layer, which helped reduce the number of misclassifications incurred by the model thereby improving the F1 scores.

Better Model: M4

Graph M3



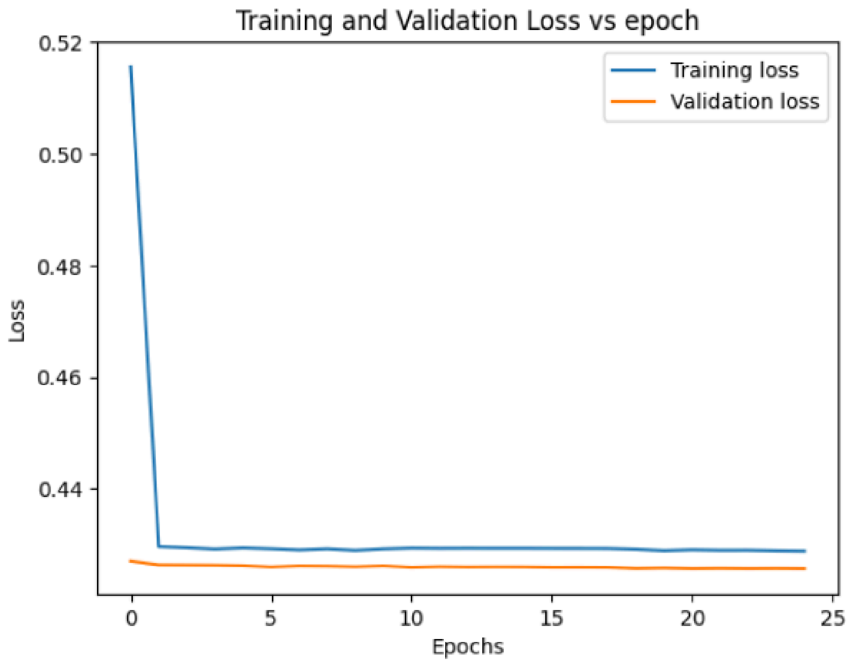
Classification Report M3

	precision	recall	f1-score	support
0	0.86	0.87	0.86	6186
1	0.22	0.20	0.21	1107
accuracy			0.77	7293
macro avg	0.54	0.54	0.54	7293
weighted avg	0.76	0.77	0.76	7293

```
F1 score for label 0: 0.8633220775044219
F1 score for label 1: 0.2085661080074488
```

```
Out[36]:
({0: 6186, 1: 1107}, {1: 1041, 0: 6252})
```

Graph M4



<u>Classification Report M4</u>

```
              precision    recall  f1-score   support

           0       0.91      0.95      0.93      6186
           1       0.66      0.50      0.57      1107

    accuracy                           0.89      7293
   macro avg       0.79      0.73      0.75      7293
weighted avg       0.88      0.89      0.88      7293

F1 score for label 0: 0.9339981006647673
F1 score for label 1: 0.5723076923076922

({0: 6186, 1: 1107}, {0: 6450, 1: 843})
```

<u>The saved models and checkpoints can be found here:</u>

https://drive.google.com/drive/folders/1qnk7xR4lvb7U65miyd4gk93tulofINYr?usp=sharing

Contributions

1. **Karan Gupta (2021258):** Model 1
2. **Raghav Sakhuja (2021274):** Model 2, Model 3, Model 4
3. **Shivesh Gulati (2021286):** Model 3, Model 4
4. **Rahul Oberoi (2021555):** Model 1, Model 2