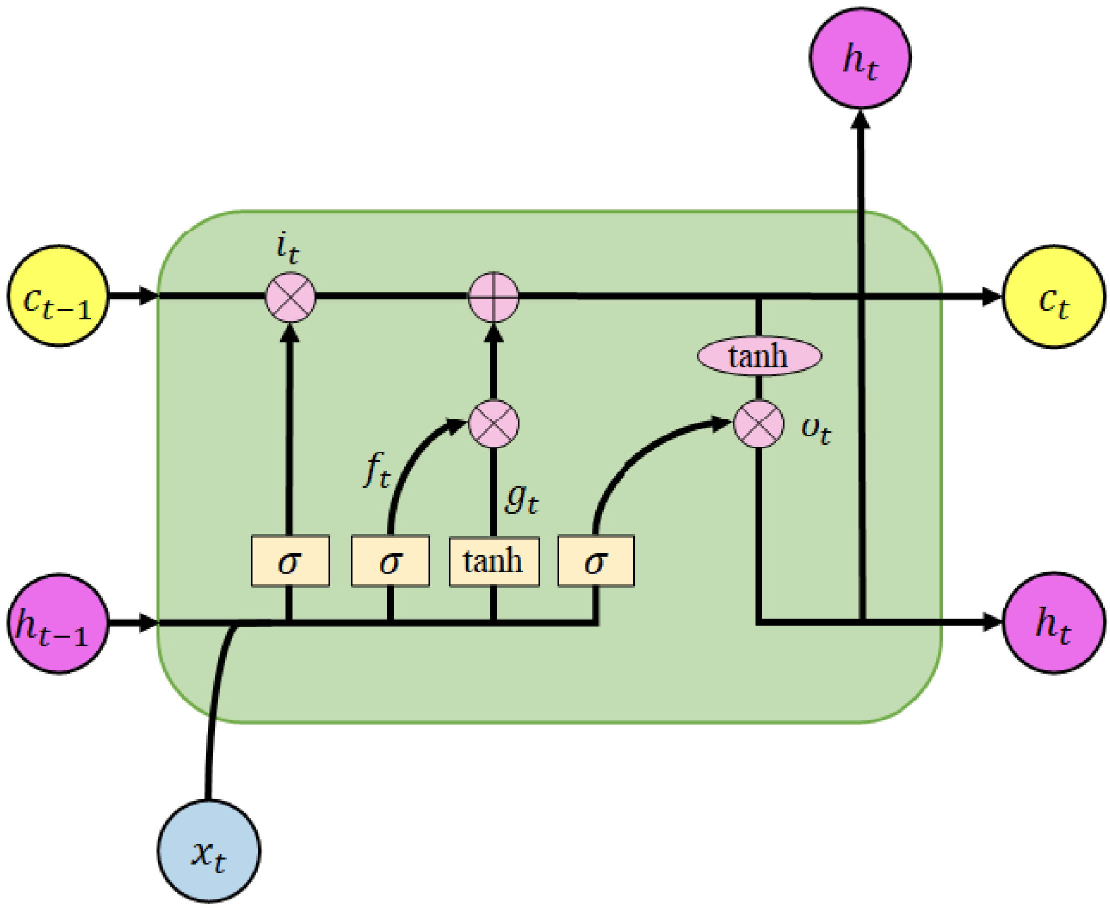
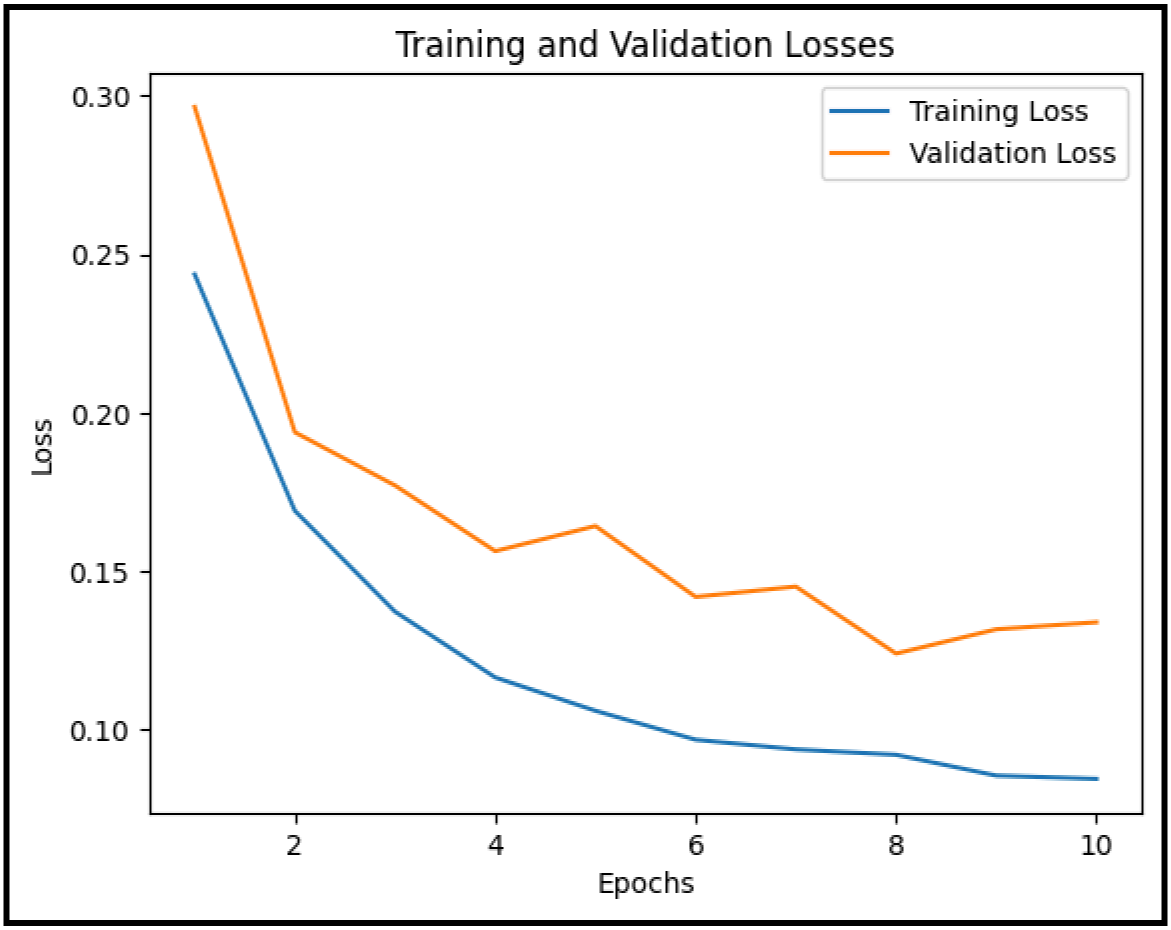
TASK-1

M1:(SBERT embeddings +LSTM)
Accuracy= `Accuracy on val data: 0.96`

Architecture:

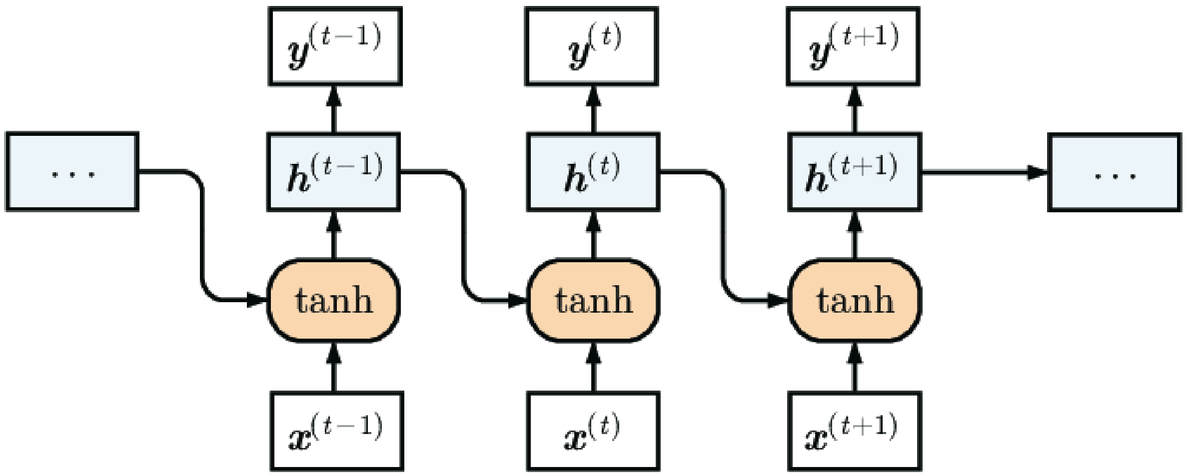


Training and validation plots

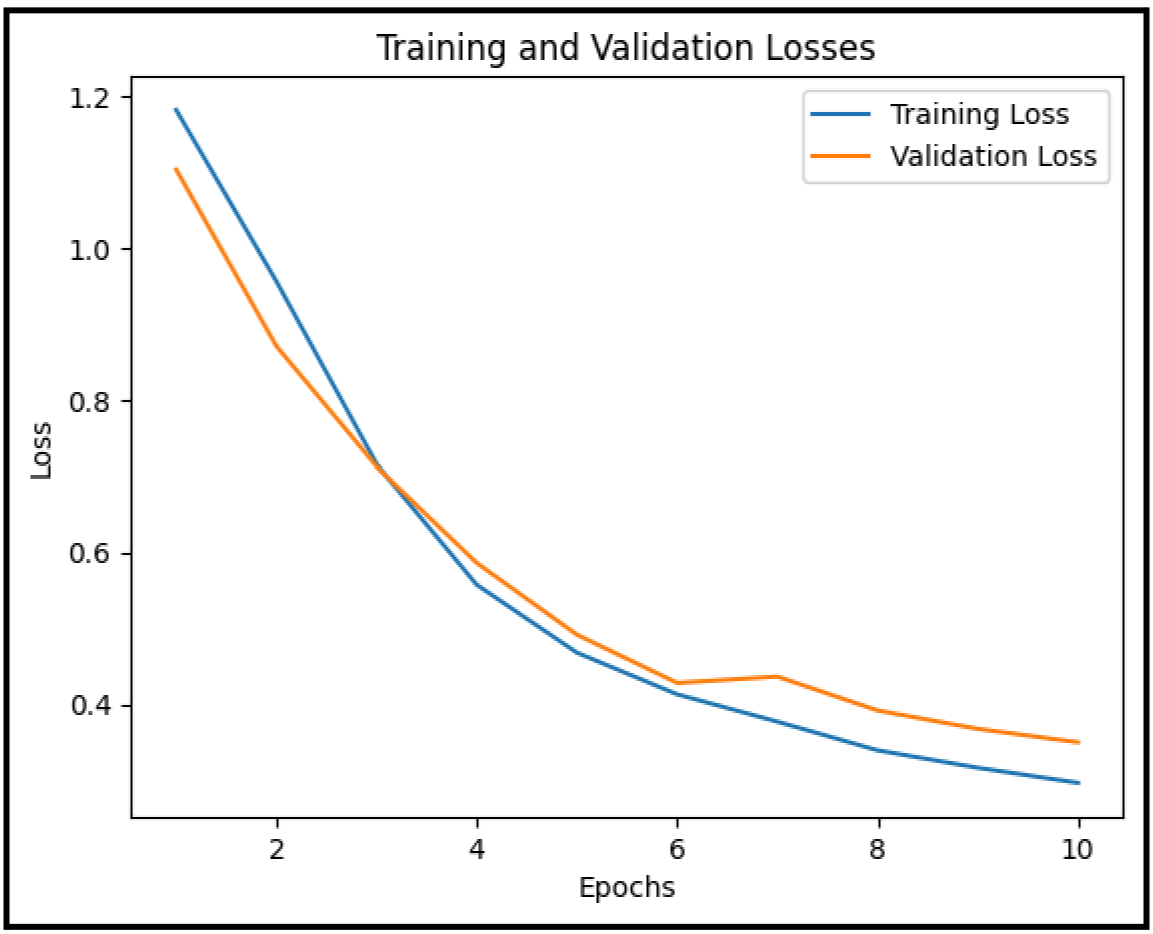


M2: (SBERT embeddings +RNN)
Accuracy: `Accuracy on val data: 0.89`

Architecture:



Plots:

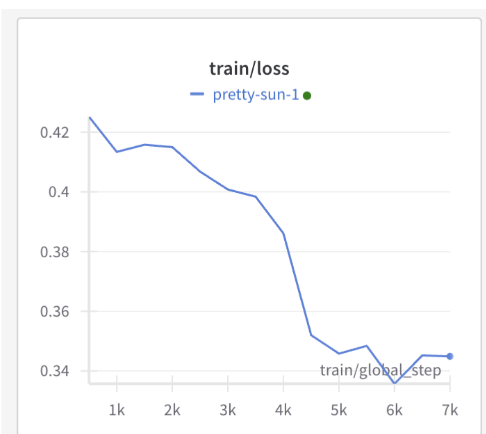


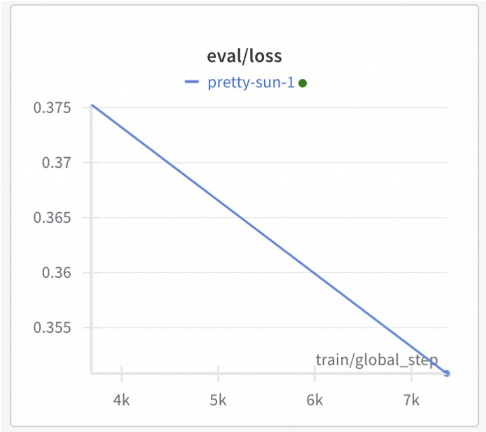
M1 performs better than M2 on the validation dataset
LSTM tends to perform better than RNN because of its ability to handle vanishing gradient problems and capture long-term dependencies.

LSTMs can consider a larger context than RNN as their memory is larger. It has gating mechanisms that help it capture relevant information and filter out noise

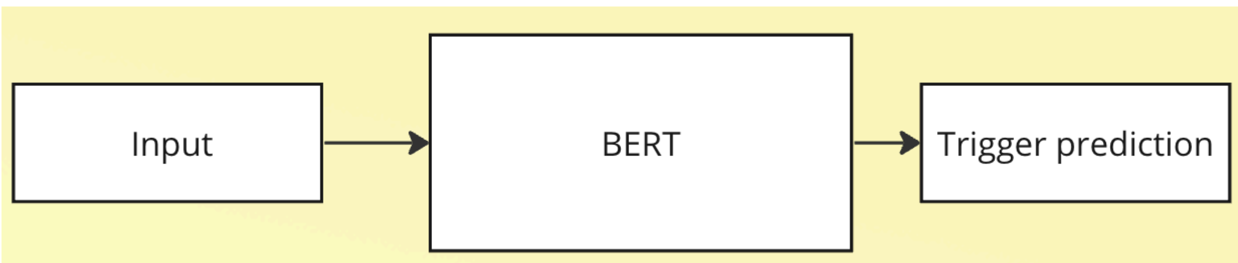
Task-2

M3: F1 score:0.414741





Architecture:



Text input (example) : "You didn't notice she was wearing different clothes?! (surprise) <sep> Oh. Ew! Ew! Ew! Ugh! Y'know what? This is too weird. (disgust)"
The text input is tokenized using DistilBERT tokenizer.

The dataset is transformed to the format with the "text" and "labels" columns. The text column contains the utterance concatenated with its emotion and the target utterance with its emotion, separated by a special token. The labels column contains 0 or 1 indicating whether the utterance is a trigger or not for the target utterance's emotion flip.
The data is preprocessed using the DistilBERT tokenizer. Pre-trained model distilbert-base-uncased is used to classify the data into trigger and non trigger.

Intuition: The intuition behind using BERT in this code is to leverage its powerful language representation capabilities for the Emotion-Flip Reasoning (EFR) task. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that has been trained on a vast amount of text data, allowing it to capture rich contextual information and semantic relationships within text sequences.

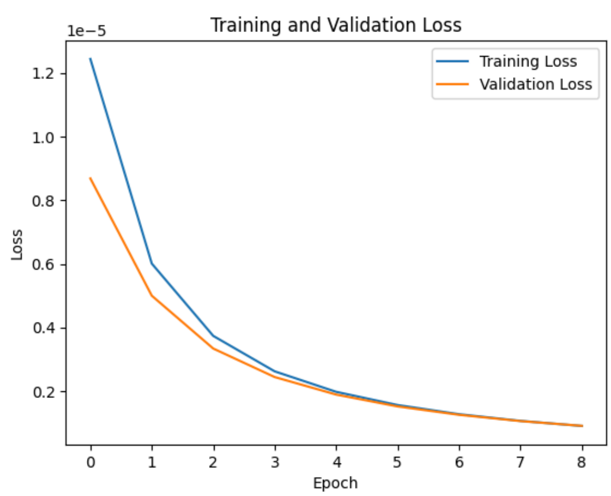
The training data is preprocessed in a way that integrates emotional information along with the utterances. Specifically, in the create_dataset function, the text input for each instance is constructed by concatenating the utterance text with its associated emotion label, separated by a special token. For example, "I'm feeling great (joy) <sep> That's awesome news (surprise)". This feature integration allows the model to learn the connections between the utterance text and the corresponding emotion, providing valuable contextual cues.

By incorporating emotion labels directly into the input text, the model can better capture the emotional state of the speakers and how it evolves throughout the conversation. This emotional context is crucial for identifying triggers that cause emotion flips, as the model can learn to associate specific utterances or patterns with changes in emotional states.

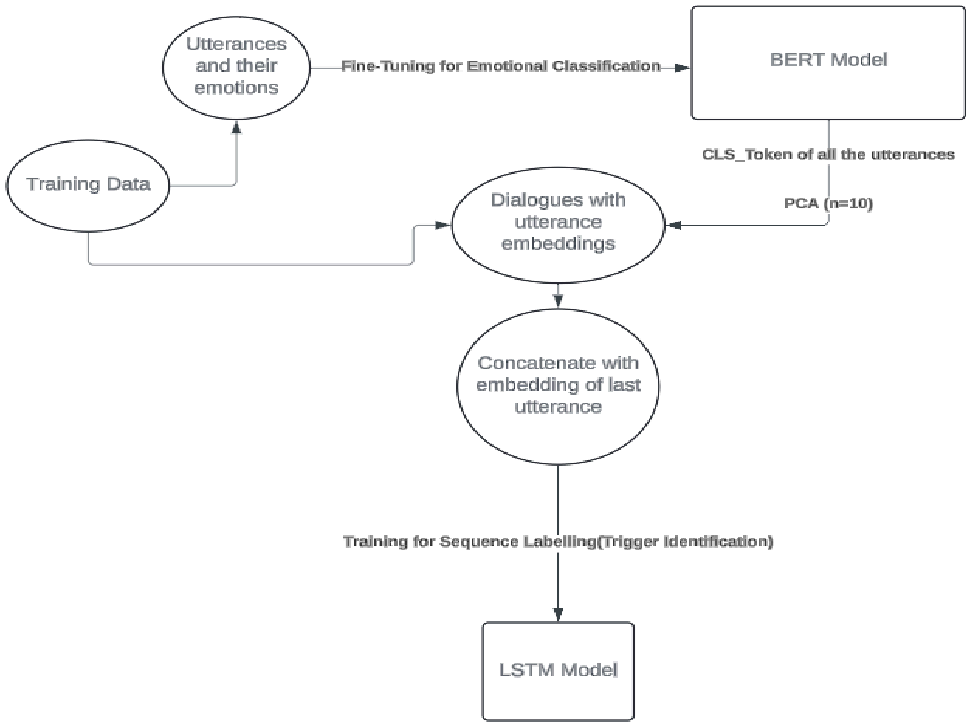
M4:

- Fine-tuned the BERT model for emotional classification.
- For each utterance in all the dialogues, predict its emotion using the BERT model.
- After this fine-tuning, passed all the utterances in the BERT model and extracted their CLS embeddings.
- Trained an LSTM model for sequence labeling (trigger or not trigger) by providing concatenated CLS embeddings of the current utterance and the utterance where emotional flip has occurred to the LSTM.
- Got an accuracy around 0.2 for trigger labeling.

Training and validation loss per epoch -



Architecture:



Intuition:

- BERT embeddings are contextual and semantically rich word embeddings. However, to make them more aligned with our task, we fine-tune the BERT model for predicting the emotions of each utterance.
- Since the trigger has to be predicted for the last utterance, we concatenate the embedding of the last utterance with the current utterance.
- In order to learn sequential context of utterances in dialogue, an LSTM model is trained on a sequence labeling task

Which model is better and why?

M3 mode (F1 score 0.4) l performs much better than M4 (F1 score 0.2). This is due to the fact that BERT model is pre-trained on a very large corpus of data. Also, both the emotions of the target utterance and current utterance are concatenated with the input, which makes the model well-informed with the context