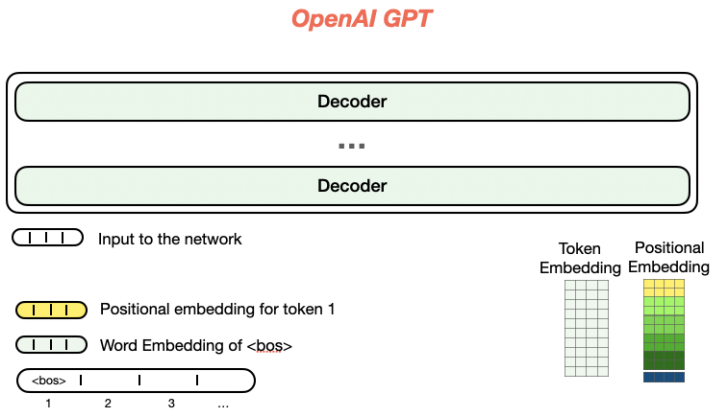
ENPM809K

Adaptation of a Transformer based chatbot with Emotion recognition for an Interactive Robot

Based on the following architecture : https://github.com/roholazandie/EmpTransfo



Author(s) :

Qamar

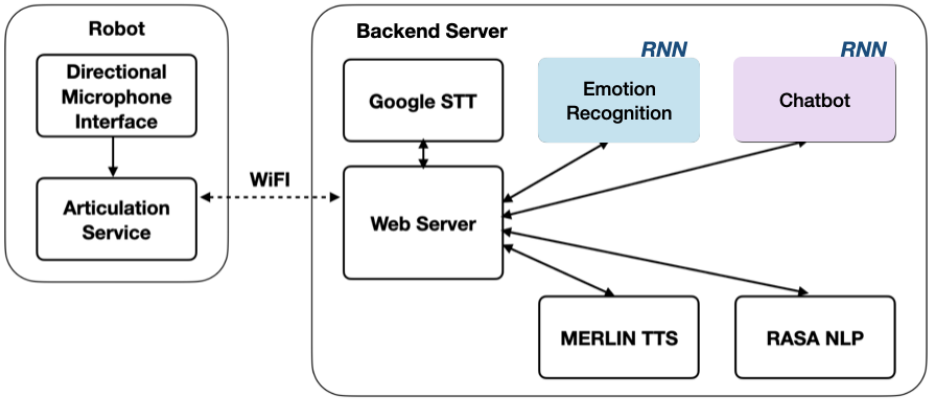
Jal

Tej Kiran

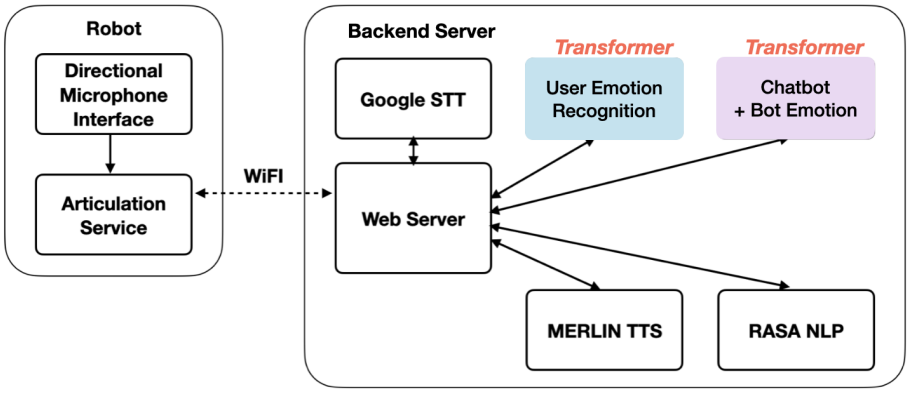
Abstract:

In this project we use a transformer based neural network to create a chatbot pipeline that understands the user emotion from text, generates a valid response and predicts an emotion score for the response. To achieve this task we will train two separate neural networks to predict user emotion and chatbot response and respective robot emotion. The AI pipeline takes in the user dialogue, predicts an user emotion and forwards the emotion and dialogue to the chatbot NN to find the response and bot's emotion. This pipeline is integrated into a real robot to demonstrate the end-to-end functionality.

Original Chatbot pipeline:



New Chatbot pipeline:



Introduction :

In the age of the global robotics revolution, it is important to enable machines to interact with humans in more expressive ways than traditional question answer systems. We have used pretrained GPT language models with different types of heads to train these two models. We have successfully trained the Emotion recognition NN with 91% validation accuracy and chatbot model with X% validation accuracy. We were able to demonstrate the context based emotions being recognized and expressed by the robot during the interaction

Related Work:

We have used the following GPT based Transformer architecture for chatbot “EmpTransfo: A Multi-head Transformer Architecture for Creating Empathetic Dialog Systems” . The model architecture we took was built by the authors of the paper based on the ‘hugging face’ pretrained model for the GPT language model. On top of GPT three different GPT compatible heads were used to train the NN. We have taken their implementation and trained two different neural networks for emotion and prediction and conversation applications.

Data:

We are using the daily_dialog data set[7] recommended by the authors of the paper for training both the neural networks.

The “***Daily-Dialog***” data set is a high quality dialogue dataset created mainly using web crawlers aimed towards English learners websites. The Data set also has been labeled manually and verified by the language experts. Unlike traditional dialogue data sets extracted from social media websites, the data set represents a more formal language and incorporates the nuances of language used by people in their everyday life.

The main goal of the data set creation is to capture information exchange and social bonding relationships in our everyday conversations. To make these relationships more concrete, each utterance of every conversation in the dataset has been explicitly labeled with the speaker's intention labels and emotion labels.

The authors of the dataset have created the categories for intents and emotions based on standard theories on these concepts. Ex: The BIG SIX Theory for key emotions. The intent categories and emotion categories are as given below.

Other Supervised Labels:
Emotions: { ANGER, DISGUST, FEAR, HAPPINESS, SADNESS, SURPRISE, OTHER}
Intents : { INFORM, QUESTIONS, DIRECTIVES, COMMISSIVE }

The overall json structure of the raw data set is as shown below

```
{       "topic" : "<attitude_and_emotion>"
        {
                "utterances":{
                        "history":   [U1, U2, .., Un ],
                        "emotion":  [E1, E2, .., En ],
                        "act":          [I1, I2, ..,I3],
                        "candidates": [U1, U2, .., Un],
                        "candidates_emotions":  [E1, E2, .., En ],
                        "candidates_acts": [I1, I2, .., I3],
                }
                {"utterances": ...}
                ....
                {"utterances": ...}
        }

        "topic" : ...
        ....
        "topic" :...
}
```

Basic Statistics:

Total Dialogues	13118
Average Speaker Turns	7.9
Average Tokens per Dialogue	114.7
Average Tokens per utterance	14.6

Feature and their distribution::

Along with the given dialogue turns, the data set provides following features for each dialogue

- *User Intention score distribution*

Inform	Questions	Directives	Commissive
46532	29428	17295	9724
45.2%	28.6%	16.8%	9.4%

- *User emotion scores distribution*

	Anger	disgust	Fear	Happiness	Sadness	Surprise
Count	1022	353	74	12885	1150	1823

The quality of the data set has been evaluated by training standard generative language models on the dataset and using the various metrics such as PPL, BLUE-N.

Dataset Evaluation metrics:

The quality of the data set has been evaluated by training standard generative language models on the dataset and using the various metrics such as PPL, BLUE-N.

PPL: It's called the Perplexity metric used for evaluating language models. It is based on the accumulation of probability of the right word given the list of the earlier words for each word in the sentence.

BLUE-N: It calculates the similarity of predicted text across a given set of reference texts. The N indicates the number of words used to calculate the similarity.
If N = 1 then BLUE-1 used one gram
If N = 2 then BLUE-2 used two gram

The reasonable highest value for N for evaluating similarity is given by the linguists as 4.

Input pre-processing:

- *Tokenization:*

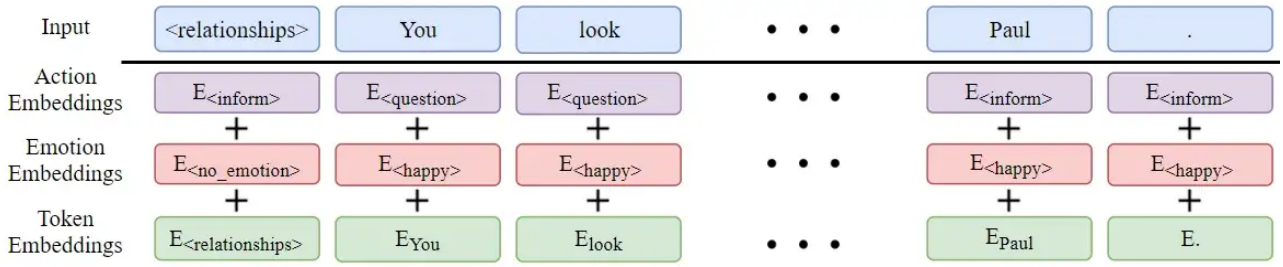
  Both neural networks take tokenized sentences as input. To tokenize the sentences, we are using the GptTokenizer module. First we generate a unique word vocabulary dictionary from the complete dataset. Then we initialize the GptTokenizer with this vocab dictionary for tokenization.

- *Adding emotions to input dialogue through tokenization:*

  Similar to how positional encoding is done in the input layer of Transformer neural network, based on the same hypothesis for information in the input layer, we are tokenizing the emotion label of the input dialogue and adding these as embedding vectors to the input sentence. By doing this the experiment assumes that input data encodes the emotion information as well.

  The emotion labels are tokenized by adding them as special labels to the vocabulary dictionary.

The input preprocessing is as shown in figure below [7]



Output of the model:

- *Emotion Recognition NN:* A 1D array of Softmax scores for all 6 Emotions
- *Chatbot NN:*
  - Choice head :- 1D array of probabilities for all possible words (40496 length) in dictionary

- Emotion head : - 1D array of Softmax scores for all 6 Emotions and NO_EMOTION label

Loss functions:

- *User Emotion Recognition NN:* CrossEntropyLoss for Emotion choice head
- *Chatbot NN:* Combined CrossEntropyLoss for both-Emotion and reply heads

Evaluation Metrics:

- *User Emotion Recognition NN:* Accuracy, Precision, Recall, Confusion Matrix
- *Chatbot NN:*
  - Reply Emotion Prediction : Accuracy
  - Chatbot reply metrics : Average_nll, Average_ppl

Methods:

Discuss your approach for solving the problems that you set up in the introduction. Why is your approach the right thing to do? Did you consider alternative approaches? You should demonstrate that you have applied ideas and skills built up during the quarter to tackling your problem of choice. It may be helpful to include figures, diagrams, or tables to describe your method or compare it with other methods.

We have taken the given architecture based on the GPT backbone model and trained two separate models to solve the described problem. We chose to go with two separate models as the user emotion generation is a sequential operation required to be performed prior to generating a response using the chatbot network.

Reasons for choosing this approach:

We decided that this approach is better than traditional architectures for two main reasons.

The First one is, by using the most modern architecture models such as transformers the network can learn to understand the context of the conversation better.

The second reason being the GPT language mode already trained on huge amounts of data, by using this as the backbone network we can achieve much better performance on most general applications such as conversational chatbots.

We have applied following skills that we learned in training the model:

Training Emotion recognition NN:

- *Understanding the Network architecture:*

  Along with the input-output layers described earlier, the complete network architecture can be summarized as below.

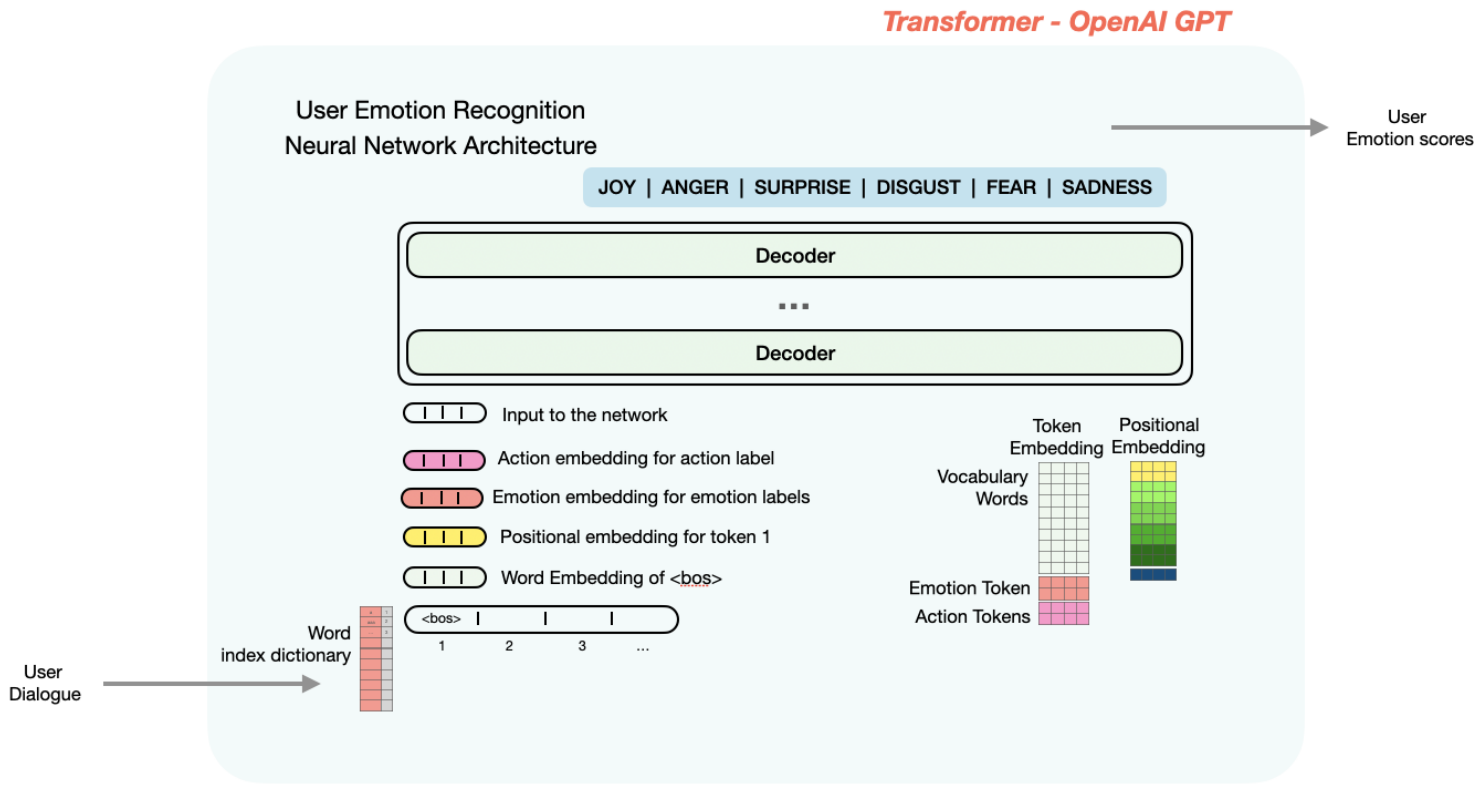


Fig: Architecture of Text Based Emotion Recognition neural network

This network takes users' dialogue as input and tries to predict the emotion level from the text.

- *Monitoring Learning rate decay and training loss :*

  Learning rate and training loss curves for the final model are as below

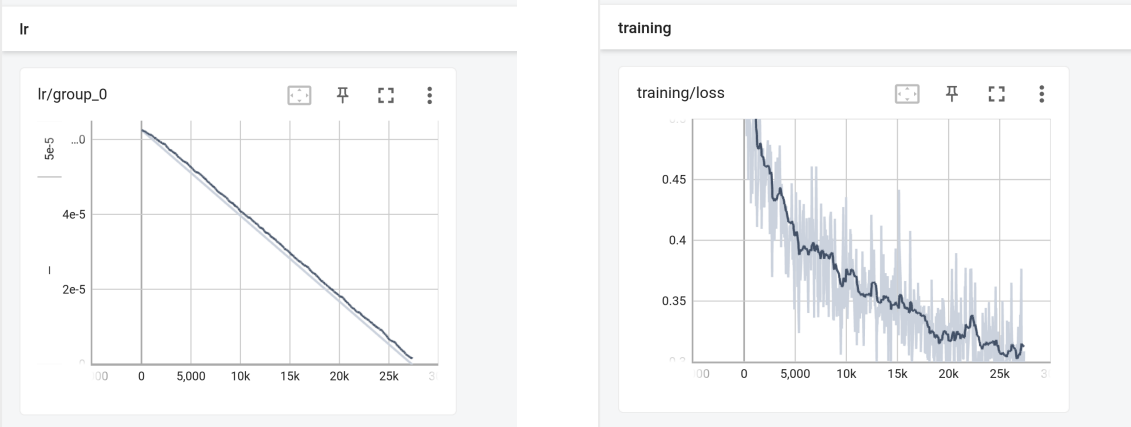
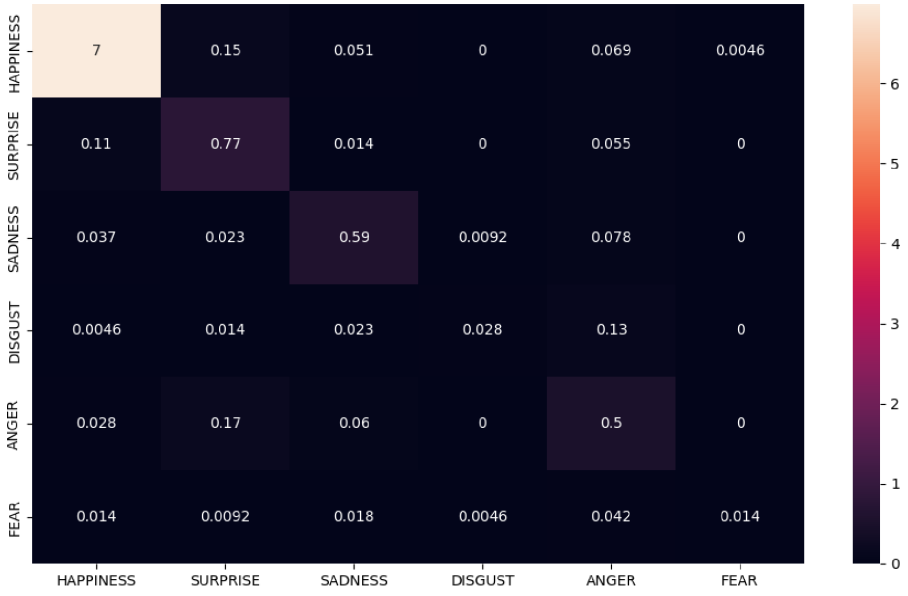


Fig: loss curves generated using tensorboard

- *Confusion matrix:*

  Confusion matrix for the final model is shown below:

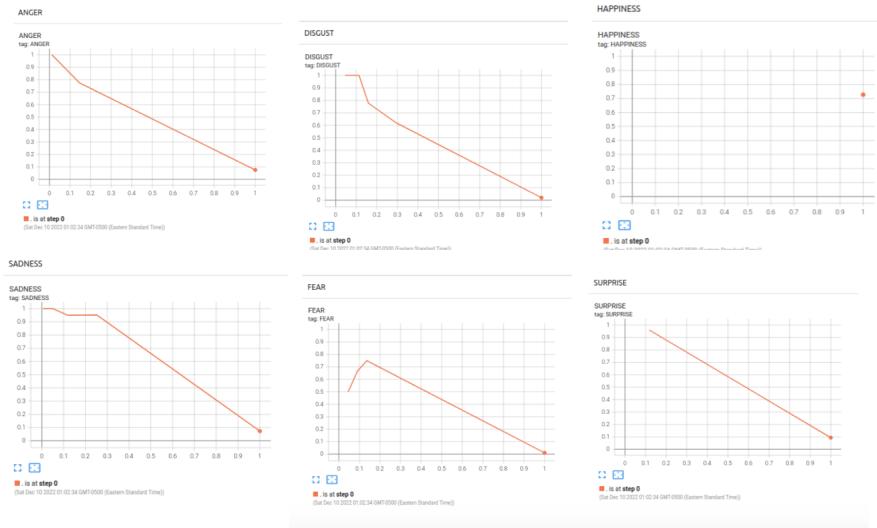


Observation: We can clearly see that because of the lower number of data samples in the categories of "FEAR" and "DISGUST", the model failed to generalize on these types of dialogues

- *Precision and Recall metrics for training time validation:*

  The precision and recall curves on validation data during training are as shown below.

- *Precision Recall Curves:*



- *Interaction results:*

```
>>> hi
HAPPINESS
>>> how are you
HAPPINESS
>>> I am very happy
HAPPINESS
>>> why are you sad?
SADNESS
>>> Look at the fireworks!
HAPPINESS
>>> This is too noisy.
ANGRY
>>> you are amazing!
HAPPINESS
>>> the movie was scary.
SURPRISE
```

- *Example of a wrong prediction*

```
>>> hi
HAPPINESS
>>> how are you
HAPPINESS
>>> I am very happy
HAPPINESS
>>> why are you sad?
SADNESS
>>> Look at the fireworks!
HAPPINESS
>>> This is too noisy.
ANGRY
>>> you are amazing!
HAPPINESS
>>> the movie was scary.
SURPRISE
>>> I am afraid of dark
HAPPINESS
```

- *Error Evaluation and Neural network interpretation using **lime framework**:*

  We have used lime frameworks to understand what part of the input sentence the network is looking at while generating the prediction.

  Below figure shows the weightage on different words that leads to the wrong prediction by the network for 1000 perturbations.

y=HAPPINESS (probability **0.325**, score **-0.896**) top features

Contribution[?]	Feature
+0.272	Highlighted in text (sum)
-0.104	
-0.163	<BIAS>
-0.288	
-0.293	I
-0.320	

i am afraid of dark

y=SURPRISE (probability **0.141**, score **-1.943**) top features

Contribution[?]	Feature
+0.508	
+0.345	
-0.389	
-0.405	Highlighted in text (sum)
-0.470	
-0.696	<BIAS>
-0.836	I

i am afraid of dark

y=DISGUST (probability **0.162**, score **-1.782**) top features

Contribution[?]	Feature
+0.751	
+0.605	
-0.360	Highlighted in text (sum)
-0.561	
-0.662	I
-0.692	
-0.863	<BIAS>

i am afraid of dark

y=ANGRY (probability **0.061**, score **-2.858**) top features

Contribution[?]	Feature
+0.627	
+0.240	
-0.533	
-0.718	<BIAS>
-0.812	Highlighted in text (sum)
-0.829	I
-0.834	

i am afraid of dark

y=SADNESS (probability **0.233**, score **-1.341**) top features

Contribution[?]	Feature
+0.419	
+0.187	
-0.168	
-0.214	I
-0.307	Highlighted in text (sum)
-0.547	<BIAS>
-0.710	

i am afraid of dark

y=FEAR (probability **0.079**, score **-2.584**) top features

Contribution[?]	Feature
+0.785	
+0.503	
-0.639	Highlighted in text (sum)
-0.670	
-0.693	<BIAS>
-0.929	
-0.941	I

i am afraid of dark

Training the Chatbot Model:

Understanding the Network architecture:

This network takes the user's dialogue and above generated emotion as input and tries to predict a reply and the emotion level of the reply.

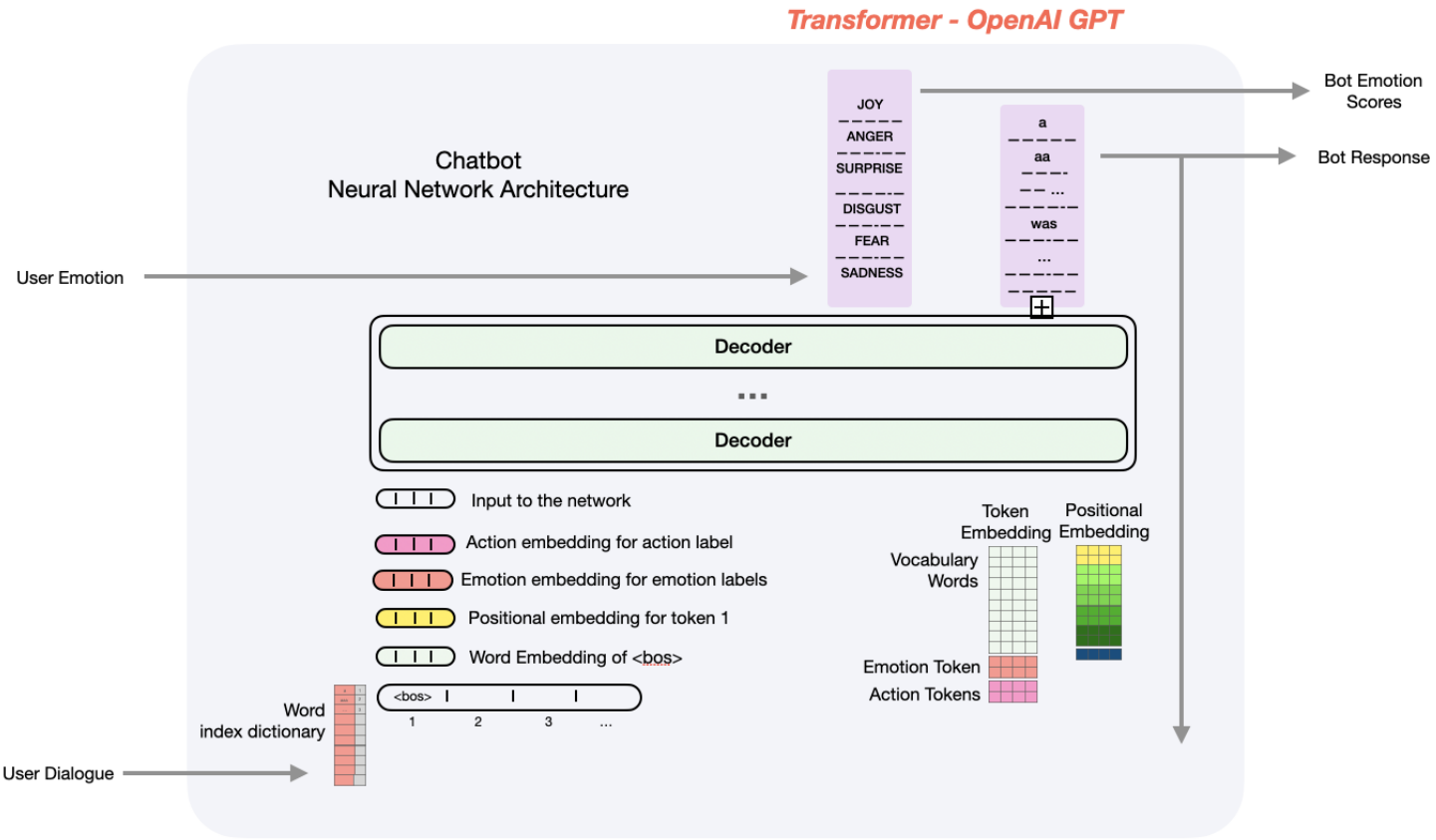
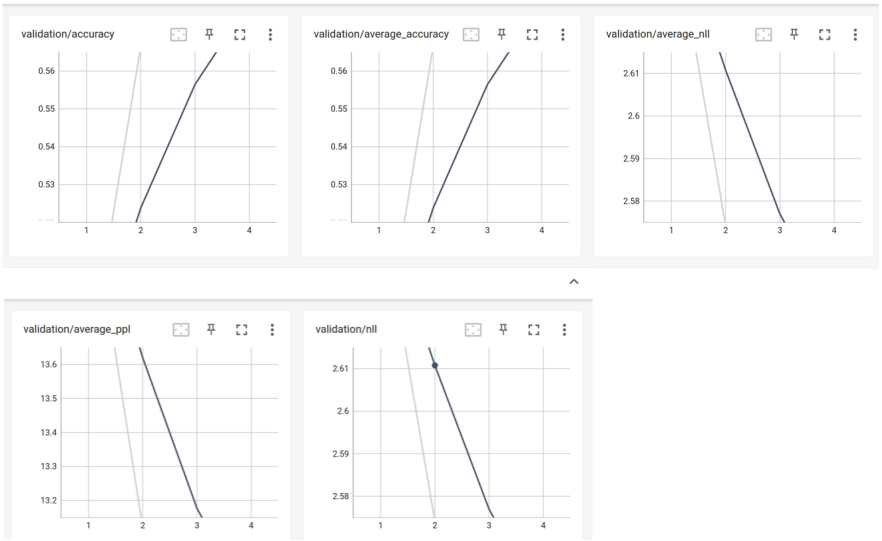


Fig: Neural network architecture of Chatbot with Emotion Generation network

Monitoring Learning rate decay and training loss :

Learning rate and training loss curves for the final model are as below



Monitoring Loss functions:

We have tried tuning different output layers by changing the loss scaling factors.The training results are as shown below.

Experiments:

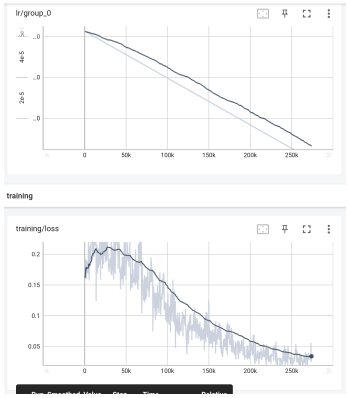
Emotion Recognition Neural Network:

Finding Number of Epochs:

Since the network began to reach 90% accuracy during the initial runs itself we have tried to over fit the network to see the minimum number of epochs required before the network begins to overfit.

By training the network for over 20 epochs we are able to reduce the training data loss from 0.3% to 0.1%. But the validation accuracy keeps fluctuating around the same value after 2-3 epochs as shown in below graphs

Overfitted network training loss and validation accuracy charts for 20 epochs:



From these charts, we can clearly see that, for the given hyper parameters 2 epochs is enough to generalize the network.

Network overfit validation:

To confirm that network overfits at 20 epochs, we tried evaluating the network using "**lime framework"** for the failure case input "I am afraid of Dark",

Confusingly, the overfitted network was able to rightly predict the output by looking at the right part of the input sentence:

```
>>> hi how are you
HAPPINESS
>>> I need to go and get dinner.
HAPPINESS
>>> But, I am afraid of dark
FEAR
>>> ■
```

This lime tools output is as shown below:

y=HAPPINESS (probability **0.020**, score **-3.861**) top features

Contribution[?]	Feature
+0.709	Highlighted in text (sum)
+0.653	
-0.395	<BIAS>
-0.549	
-1.055	
-1.257	
-1.967	I

i am afraid of dark

y=DISGUST (probability **0.000**, score **-7.601**) top features

Contribution[?]	Feature
-0.212	
-0.287	
-0.511	<BIAS>
-1.198	
-1.265	Highlighted in text (sum)
-1.735	
-2.393	I

i am afraid of dark

y=SURPRISE (probability **0.001**, score **-7.468**) top features

Contribution[?]	Feature
+0.198	
-0.250	
-0.481	<BIAS>
-0.552	
-0.828	
-2.600	I
-2.956	Highlighted in text (sum)

i am afraid of dark

y=ANGRY (probability **0.000**, score **-9.069**) top features

Contribution[?]	Feature
-0.184	
-0.194	
-0.515	<BIAS>
-1.274	
-1.378	
-2.310	I
-3.214	Highlighted in text (sum)

i am afraid of dark

y=SADNESS (probability **0.023**, score **-3.719**) top features

Contribution[?]	Feature
+1.026	
+0.130	
-0.347	Highlighted in text (sum)
-0.553	<BIAS>
-0.949	
-1.317	I
-1.708	

i am afraid of dark

y=FEAR (probability **0.956**, score **4.208**) top features

Contribution[?]	Feature
+2.573	
+2.095	I
+1.521	Highlighted in text (sum)
+0.093	
-0.350	<BIAS>
-0.683	
-1.041	

i am afraid of dark

Now, we wanted to reconfirm whether the 20_epochs network has generalized better than the 2 epochs network by understanding the meaning of “am afraid of dark”.

For this, we went back to the training set of “daily dialog “ dataset and tried looking for the sentence “I am afraid of the dark”. To our surprise, we were able to find the following sentence in the training data set.

Screen show of training data set containing the sentence ”I’m afraid of the darkness:

```
351   <fear>
352   Oh , God . It ' s late . I ' m afraid I have to leave .
353   <fear>
354   Oh , God . It ' s late . I ' m afraid I have to leave .
355   <fear>
356   It must be a haunted house . Are you frightened ?
357   <fear>
358   I'm afraid of the darkness .
359   <fear>
360   I'm afraid of the darkness .
361   <fear>
362   You think it's funny . I'm terrified .
363   <fear>
```

From this we conclude that the overfitted network predicted the right label by memorizing the input sentence from the dataset.

Chatbot Neural Network:

Because of available GPU limitations we could not run many experiments on this network. As seen below the network took ~3 hours for each epoch.

```
Epoch [1/10]: [46791/46791] 100%|, loss=0.382 [3:12:08<00:00]
INFO:ignite.engine.engine.Engine:Engine run complete. Time taken: 00:36:08.694
Epoch [1/10]: [46791/46791] 100%|, loss=0.382 [3:12:08<00:00]
INFO:ignite.engine.engine.Engine:Epoch[1] Complete. Time taken: 03:12:11.990
Epoch [2/10]: [46791/46791] 100%|, loss=0.296 [2:50:25<00:00]
```

• Limitations :

Some major limitations of our approach are as follows:

1) Two separate networks are required to predict the user emotion and chatbot response.
2) Since the user emotion goes as an input to the chatbot network, the prediction pipeline has to execute two neural networks at run time which might not be computationally efficient.
3) Finally, since we are using a larger model, such as GPT as the backbone, it might not be feasible to deploy them directly on the hardware for offline usage.

• Conclusion:

In this project, we have successfully created an emotion recognition and response generation pipeline for an interactive robot and performed various experiments to evaluate their performance and demonstrated its application on a real robot.

• Contributions of various tasks:

- **Study**
- **Understanding the dataset**
- **Understanding network inputs and outputs**
- **Initial training**
- **Creation network of inference scripts**
- **Creating chatbot pipeline interfaces and deployment on the robot**
- **Setting up google collab environment**
- **Creating collab compatible scripts for training**
- **Experiments:**
- **Understanding loss curves:**
- **Understanding network output**
- **Evaluating the output of overfitted network**

- **Interpreting the network using the Lime framework**
- **Documentation**

References:

[1] https://arxiv.org/abs/1706.03762
[2] https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
[3] https://huggingface.co/docs/transformers/model_doc/openai-gpt
[4] https://medium.com/voice-tech-podcast/emptransfo-how-to-create-a-chatbot-that-understands-emotion-97071de0d0f4
[5] https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)
[6] https://arxiv.org/abs/1710.03957
[7] https://github.com/roholazandie/EmpTransfo