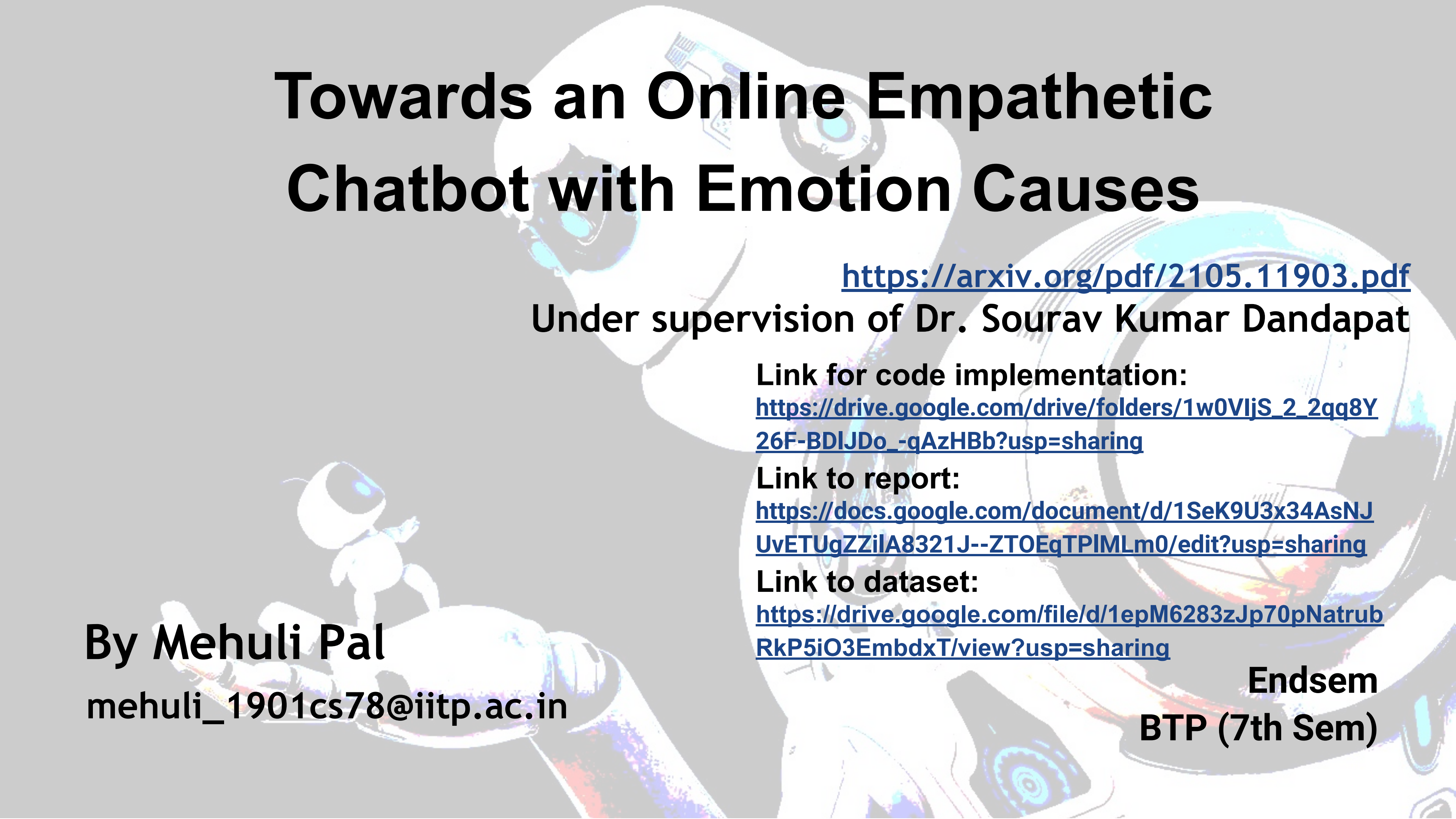


Towards an Online Empathetic Chatbot with Emotion Causes

https://arxiv.org/pdf/2105.11903.pdf

Under supervision of Dr. Sourav Kumar Dandapat

Link for code implementation:
https://drive.google.com/drive/folders/1w0VIjS_2_2qq8Y26F-BDlJDo_-qAzHBb?usp=sharing

Link to report:
https://docs.google.com/document/d/1SeK9U3x34AsNJUvETUgZZilA8321J--ZTOEqTPlMLm0/edit?usp=sharing

Link to dataset:
https://drive.google.com/file/d/1epM6283zJp70pNatrubRkP5iO3EmbdxT/view?usp=sharing

By Mehuli Pal
mehuli_1901cs78@iitp.ac.in

Endsem
BTP (7th Sem)

Contents

- What is Empathy?
- Literature review
- Limitations of existing models
- Introduction to EMMA
- Understanding approach
- Understanding architecture
- Exploring GPT-2
- Potential models
- Mathematical formula
- Negative log-likelihood loss
- Adam optimizer
- Chinese dataset results
- Preparing English dataset
- Annotating English dataset
- English dataset results
- Future scope

What is Empathy?

Empathy is the ability to emotionally understand what other people feel, see things from their point of view, and imagine yourself in their place. Essentially, it is putting yourself in someone else's position and feeling what they are feeling.

Literature Review

- The first chatbot was ELIZA, constructed in 1966. Its ability to communicate was limited, but it was a source of inspiration for the subsequent development of other chatbots.
- In 1972, PARRY appeared. It is considered more advanced than ELIZA as it is supposed to have a “personality” and a better controlling structure.
- The term Chatterbot was first mentioned in 1991. It was a TINYMUD (multiplayer real-time virtual world) artificial player, whose primary function was to chat.
- In 2001, there was a real evolution in chatbot technology with the development of SmarterChild, which was available on Messengers like America Online and Microsoft.
- Apple Siri, IBM Watson, Google Assistant, Microsoft Cortana, and Amazon Alexa are the most popular voice assistants of today.

Limitations of Existing Models

- Focus on controlling the response contents to align with a specific *emotion class*
- Unable to understand or concern the feelings and experience of others
- Tend to produce responses that are rarely empathetic
- But empathy plays a vital role for amicable social conversation and trustful social bonding

Limitations of Existing Models

Focus on controlling the response contents to align with a specific ***emotion class***

Turn	Utterance	Strategy & Cause
U1	I'm upset.	None
S1	Everything will be OK.	None

Based on - **EMOTION CLASS**

Introduction to EMMA

- Online **Em**pathetic chatbot based on the user e**m**otion c**a**uses
- Learns the causes that evoke the users’ emotion for empathetic responding, a.k.a. *emotion causes*
- Not only understand what is being discussed, but also acknowledge the implied feelings of the conversation and respond appropriately

Based on - **EMOTION CLASS** + **EMOTION CAUSES**

Turn	Utterance	Strategy & Cause
U1	I’m upset.	None
S1	Everything will be OK.	None

Existing approach - EMOTION CLASS

Turn	Utterance	Strategy & Cause
U1	I’m upset.	None
S1	Sorry to hear that. What happened?	Effective questioning
U2	We ***broke up***.	Emotion cause
S2	Oh dear, it must be hurt. Did you argue for something?	Active listening

EMMA - EMOTION CLASS + EMOTION CAUSES

Approach

1. Starts a conversation

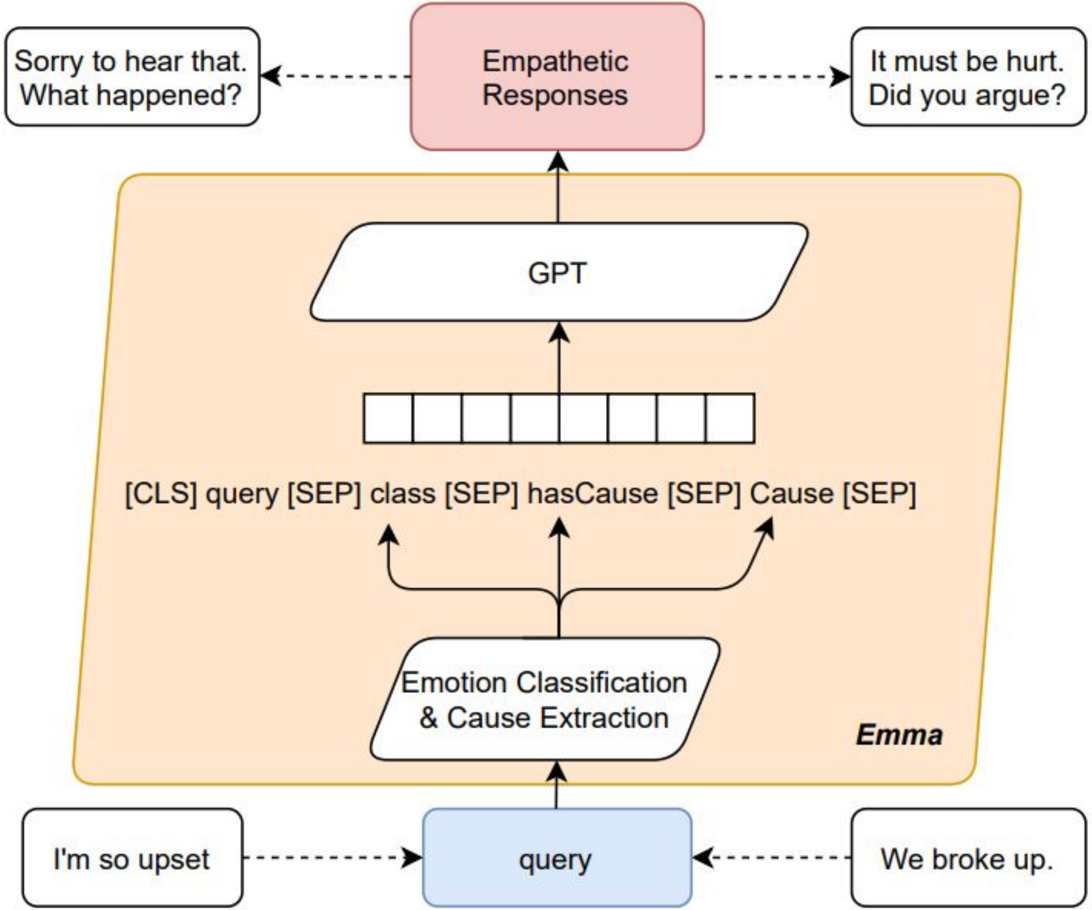
2. Detects user emotion class

3. Recognizes emotion causes

4. If no emotion cause is detected, Emma directs users to self-disclose more based on ***effective questioning*** and ***active listening***

5. Produces empathetic responses based on the ***conversation history***, detected ***emotion class*** and ***emotion causes***

Architecture



GPT-2

Generative Pretrained Transformer 2 is an autoregressive language model that uses deep learning to produce human-like text.

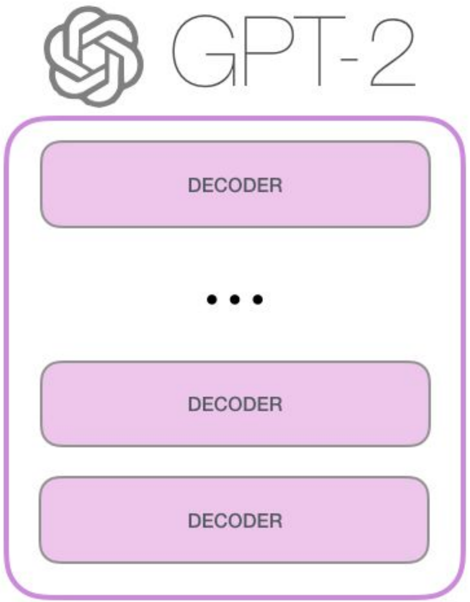
Given prompt 1: *The dog on the ship ran*

Generated prompt 1: *The dog on the ship ran* ***off, and the dog was found by the crew.***

Given prompt 2: *The motor on the ship ran*

Generated prompt 2: *The motor on the ship ran* ***at a speed of about 100 miles per hour.***

Potential Models



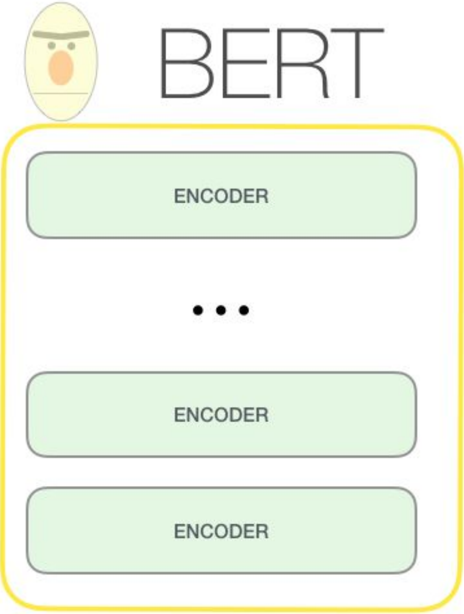
GPT-2

DECODER

• • •

DECODER

DECODER



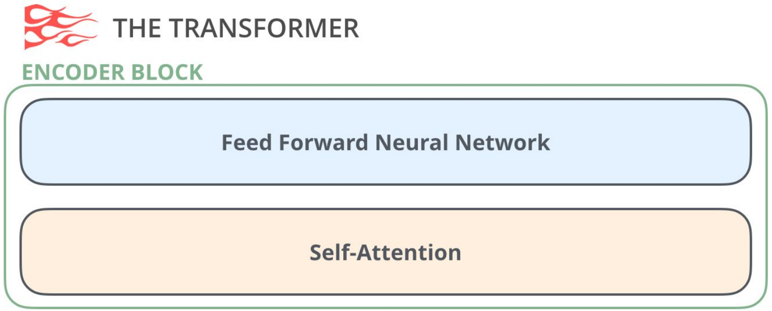
BERT

ENCODER

• • •

ENCODER

ENCODER



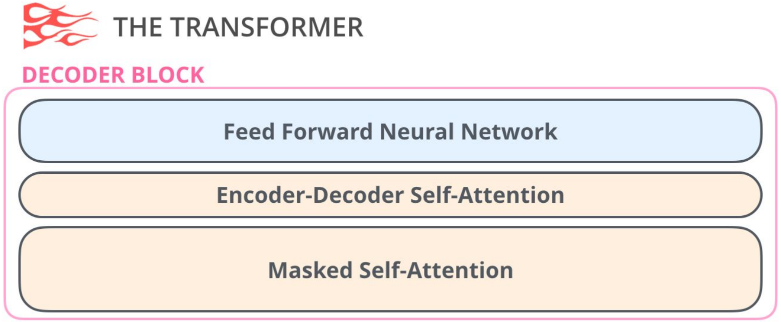
THE TRANSFORMER

ENCODER BLOCK

Feed Forward Neural Network

Self-Attention

robot	must	obey	orders	<eos>	<pad>	...	<pad>
1	2	3	4	5	6		512



THE TRANSFORMER

DECODER BLOCK

Feed Forward Neural Network

Encoder-Decoder Self-Attention

Masked Self-Attention

Input

<s>	robot	must	obey				
1	2	3	4	5	6		512

Mathematical Formula

$$P_Y = \prod_{i=1}^{T} P(y_t \mid y_{0:t-1}, X, H, L, C)$$

Query : $X = \{x1, \cdots, xN\}$
H : History conversations
L : Emotion class label
C : Emotion causes
Response : $Y = \{y1, \cdots, yT\}$

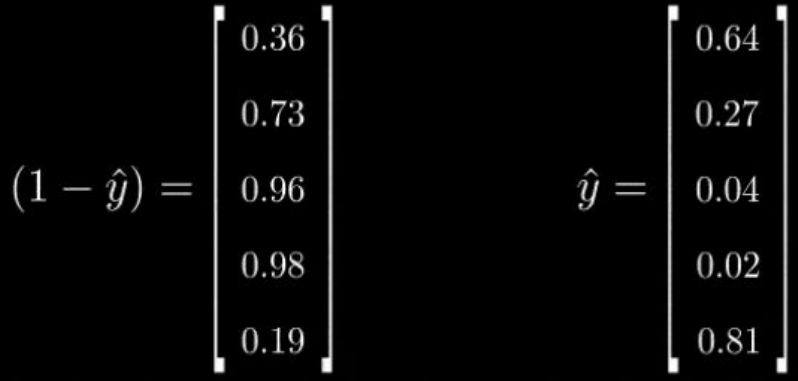
[CLS] [speaker1] query1 [speaker2] response1 [speaker1] query2 [SEP] label [SEP] hasCause [SEP] Cause [SEP]

Our task is to learn a ‘response generation model’ via ‘maximum likelihood estimation’

Negative Log-Likelihood (NLL) Loss

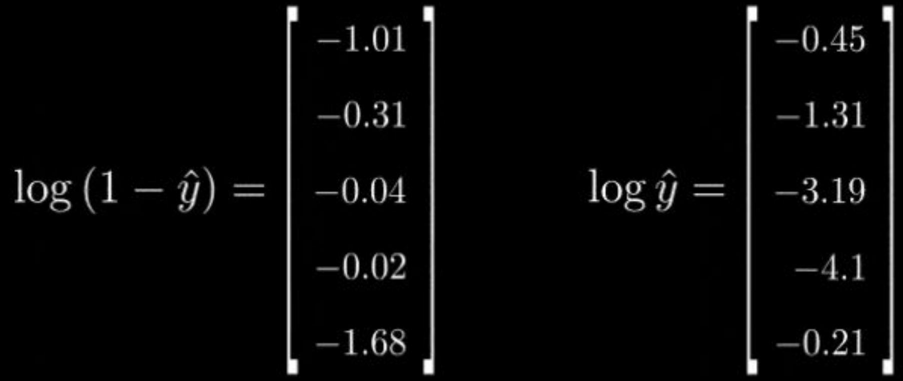
- Negative log-likelihood minimization is a proxy problem to the problem of ***maximum likelihood estimation***.
- Loss function is given by ***L(ˆy, y)***, where ***ˆy*** represents the ***predicted output*** and ***y*** represents ***true output***.
- The training objective is then to minimize the loss across the different training examples.

$$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_i y_i \log(\hat{y}_i)$$



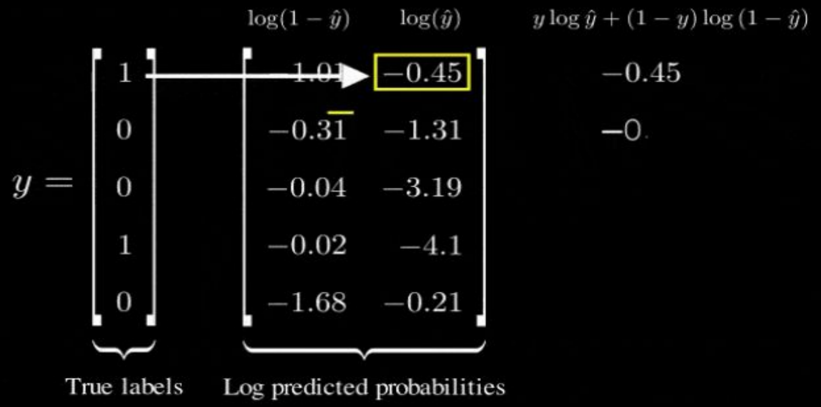
$$(1-\hat{y}) = \begin{bmatrix} 0.36 \\ 0.73 \\ 0.96 \\ 0.98 \\ 0.19 \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} 0.64 \\ 0.27 \\ 0.04 \\ 0.02 \\ 0.81 \end{bmatrix}$$

Step-I



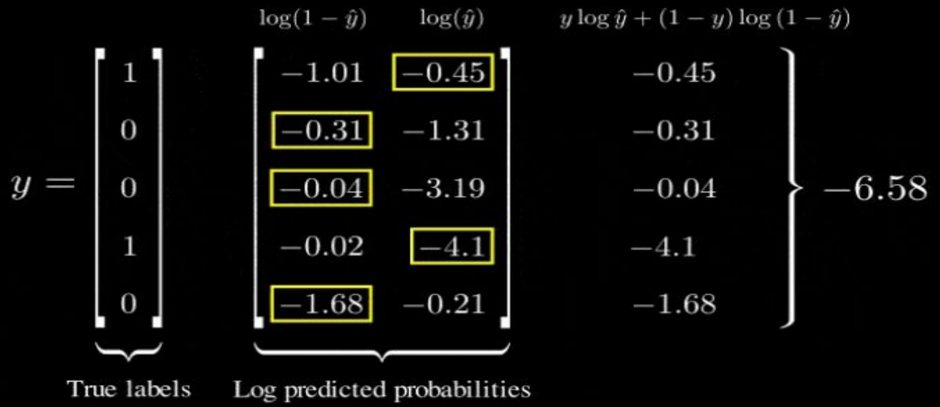
$$\log(1-\hat{y}) = \begin{bmatrix} -1.01 \\ -0.31 \\ -0.04 \\ -0.02 \\ -1.68 \end{bmatrix} \qquad \log\hat{y} = \begin{bmatrix} -0.45 \\ -1.31 \\ -3.19 \\ -4.1 \\ -0.21 \end{bmatrix}$$

Step-II



$$y = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{True labels}} \underbrace{\begin{array}{cc} \log(1-\hat{y}) & \log(\hat{y}) \\ \hline -1.01 & \boxed{-0.45} \\ -0.31 & -1.31 \\ -0.04 & -3.19 \\ -0.02 & -4.1 \\ -1.68 & -0.21 \end{array}}_{\text{Log predicted probabilities}} \quad \begin{array}{c} y\log\hat{y} + (1-y)\log(1-\hat{y}) \\ \hline -0.45 \\ -0. \\ \\ \\ \\ \end{array}$$

Step-III



$$y = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{True labels}} \underbrace{\begin{array}{cc} \log(1-\hat{y}) & \log(\hat{y}) \\ \hline -1.01 & \boxed{-0.45} \\ \boxed{-0.31} & -1.31 \\ \boxed{-0.04} & -3.19 \\ -0.02 & \boxed{-4.1} \\ \boxed{-1.68} & -0.21 \end{array}}_{\text{Log predicted probabilities}} \quad \left.\begin{array}{c} y\log\hat{y} + (1-y)\log(1-\hat{y}) \\ \hline -0.45 \\ -0.31 \\ -0.04 \\ -4.1 \\ -1.68 \end{array}\right\} -6.58$$

Step-IV

Adam Optimizer

Adaptive Moment Estimation is an optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient.

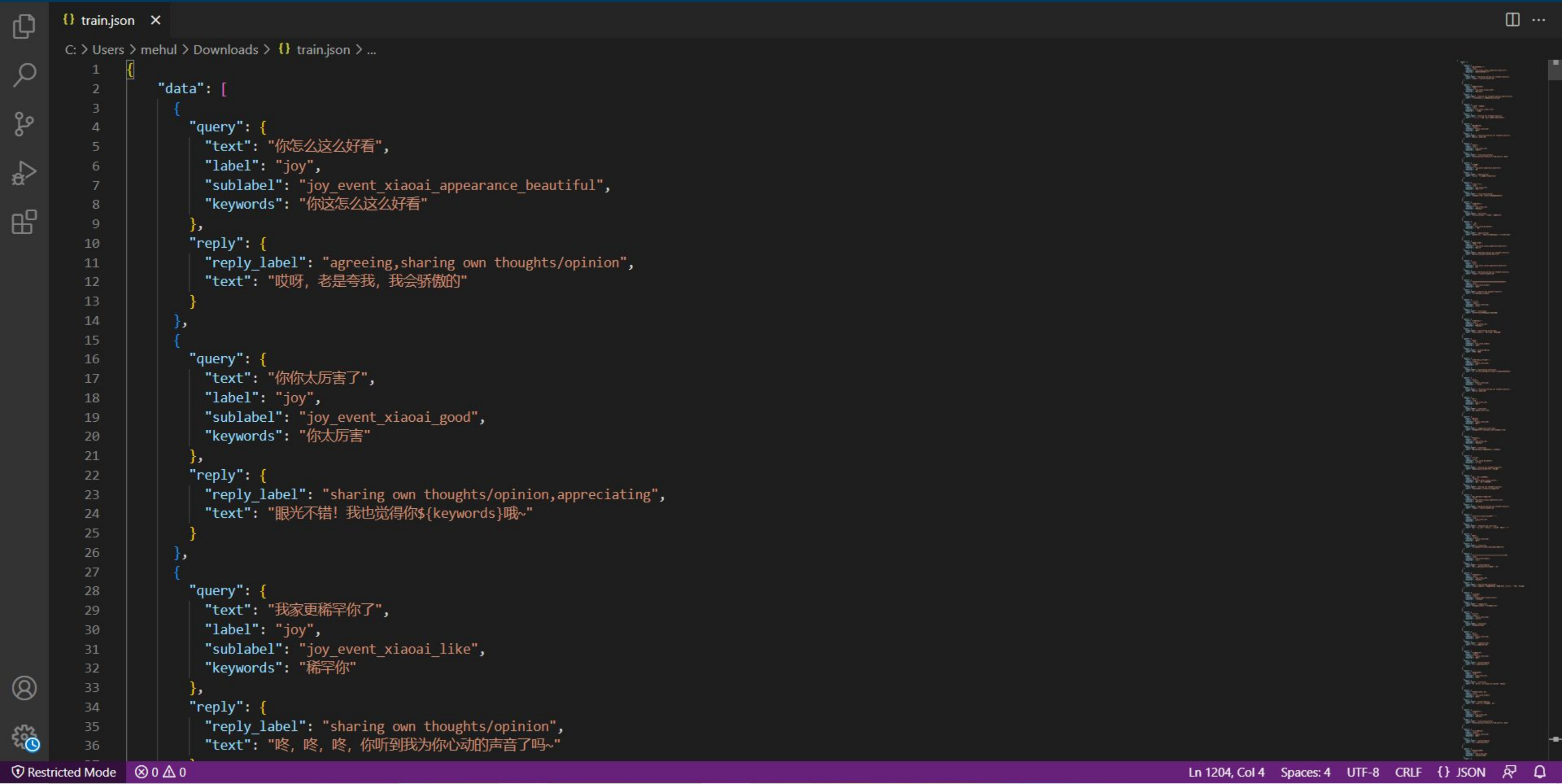
Adam optimizer involves a combination of two gradient descent methodologies:

- Gradient Descent with Momentum
- Root Mean Square Propagation (RMSP)

Update rule for Adam optimizer (w = weight):

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

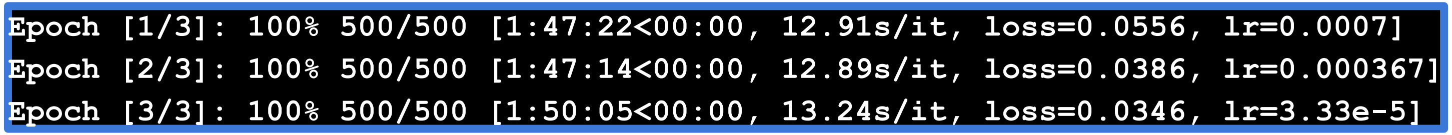
Training on XiaoMi EMMA Chinese Dataset



```
{
    "data": [
      {
        "query": {
          "text": "你怎么这么好看",
          "label": "joy",
          "sublabel": "joy_event_xiaoai_appearance_beautiful",
          "keywords": "你这怎么这么好看"
        },
        "reply": {
          "reply_label": "agreeing,sharing own thoughts/opinion",
          "text": "哎呀，老是夸我，我会骄傲的"
        }
      },
      {
        "query": {
          "text": "你你太厉害了",
          "label": "joy",
          "sublabel": "joy_event_xiaoai_good",
          "keywords": "你太厉害"
        },
        "reply": {
          "reply_label": "sharing own thoughts/opinion,appreciating",
          "text": "眼光不错！我也觉得你${keywords}哦~"
        }
      },
      {
        "query": {
          "text": "我家更稀罕你了",
          "label": "joy",
          "sublabel": "joy_event_xiaoai_like",
          "keywords": "稀罕你"
        },
        "reply": {
          "reply_label": "sharing own thoughts/opinion",
          "text": "咚，咚，咚，你听到我为你心动的声音了吗~"
```

Chinese Dataset Training Parameters & Results

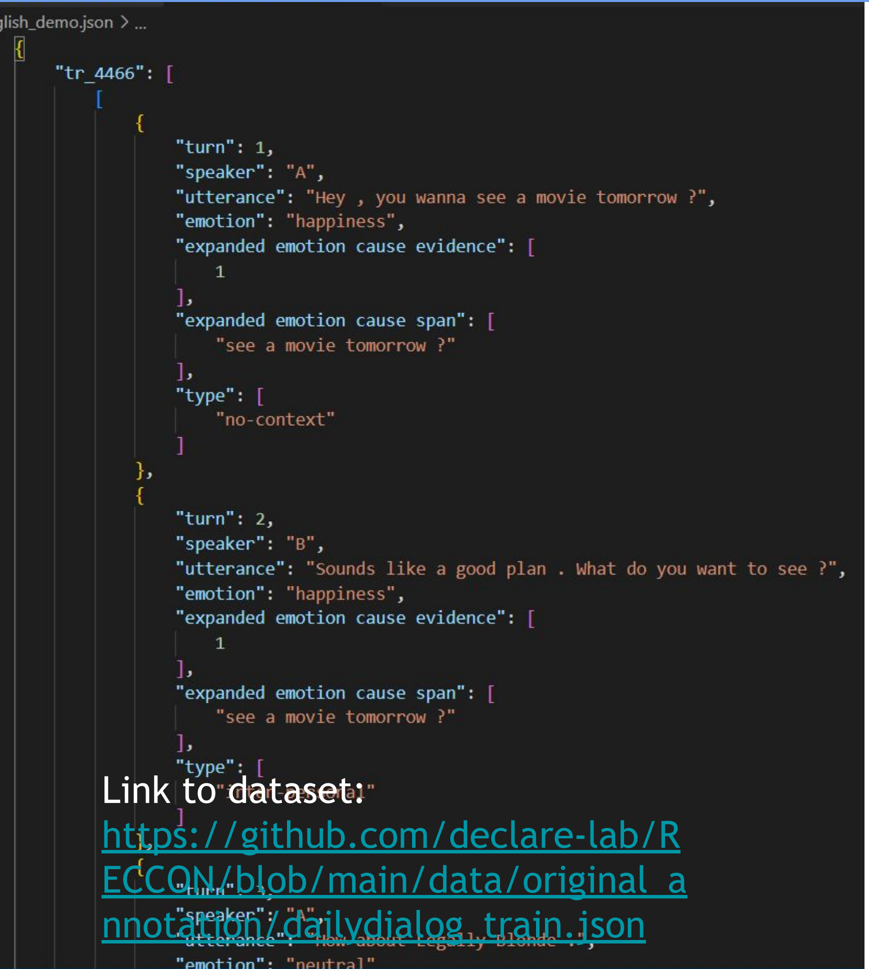
Language: Chinese
Number of records: 80,000
Model: bert-base-uncased
Learning rate: 0.001
Number of epochs: 3
Training batch size: 16
Validation batch size: 16



```
Epoch [1/3]: 100% 500/500 [1:47:22<00:00, 12.91s/it, loss=0.0556, lr=0.0007]
Epoch [2/3]: 100% 500/500 [1:47:14<00:00, 12.89s/it, loss=0.0386, lr=0.000367]
Epoch [3/3]: 100% 500/500 [1:50:05<00:00, 13.24s/it, loss=0.0346, lr=3.33e-5]
```

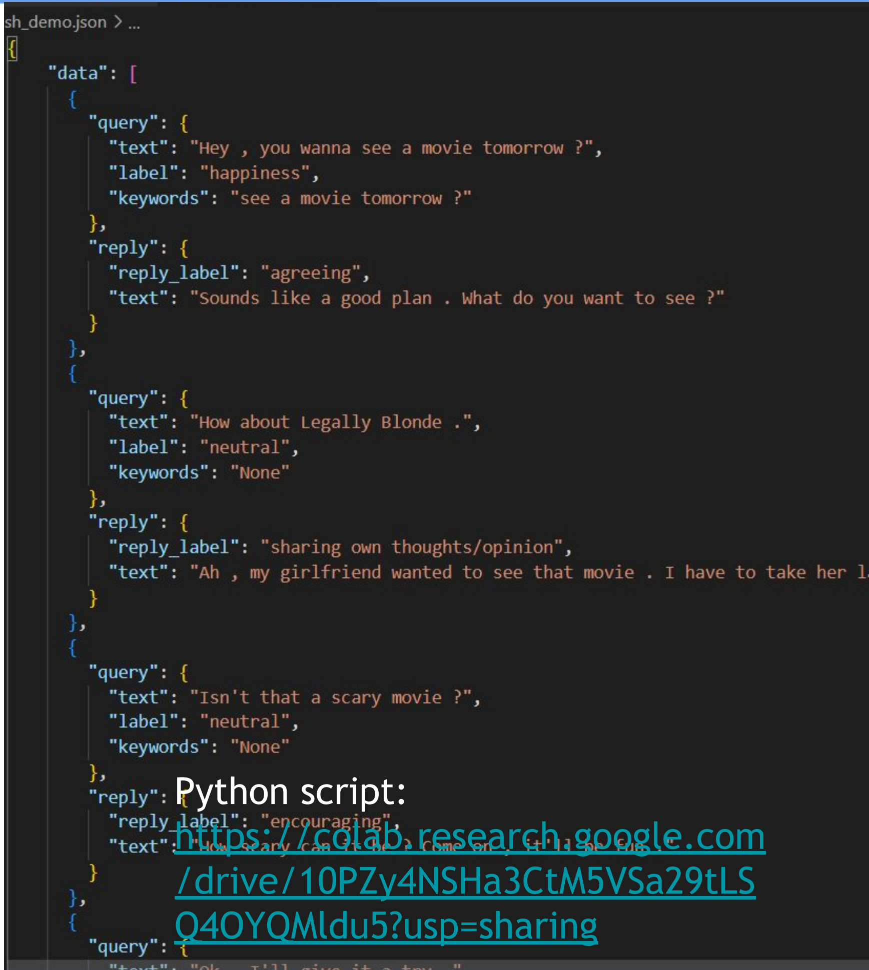
Reference: https://github.com/XiaoMi/emma/tree/master/data/raw

Preparing English Dataset using Python Script



```
glish_demo.json > ...
{
    "tr_4466": [
        [
            {
                "turn": 1,
                "speaker": "A",
                "utterance": "Hey , you wanna see a movie tomorrow ?",
                "emotion": "happiness",
                "expanded emotion cause evidence": [
                    1
                ],
                "expanded emotion cause span": [
                    "see a movie tomorrow ?"
                ],
                "type": [
                    "no-context"
                ]
            },
            {
                "turn": 2,
                "speaker": "B",
                "utterance": "Sounds like a good plan . What do you want to see ?",
                "emotion": "happiness",
                "expanded emotion cause evidence": [
                    1
                ],
                "expanded emotion cause span": [
                    "see a movie tomorrow ?"
                ],
                "type": [
                    "inter-personal"
                ]
            },
            {
                "turn": 3,
                "speaker": "A",
                "utterance": "How about Legally Blonde .",
                "emotion": "neutral"
```

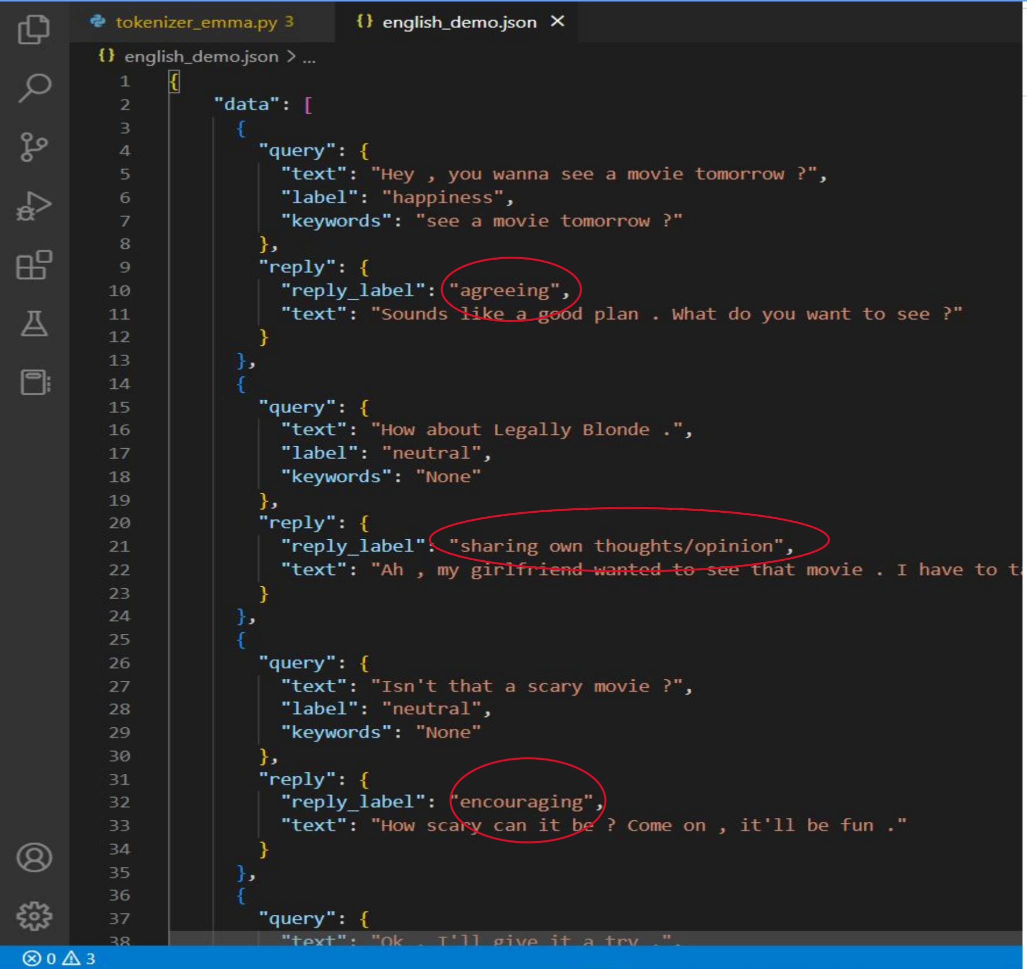
Link to dataset:
https://github.com/declare-lab/RECCON/blob/main/data/original_annotation/dailydialog_train.json



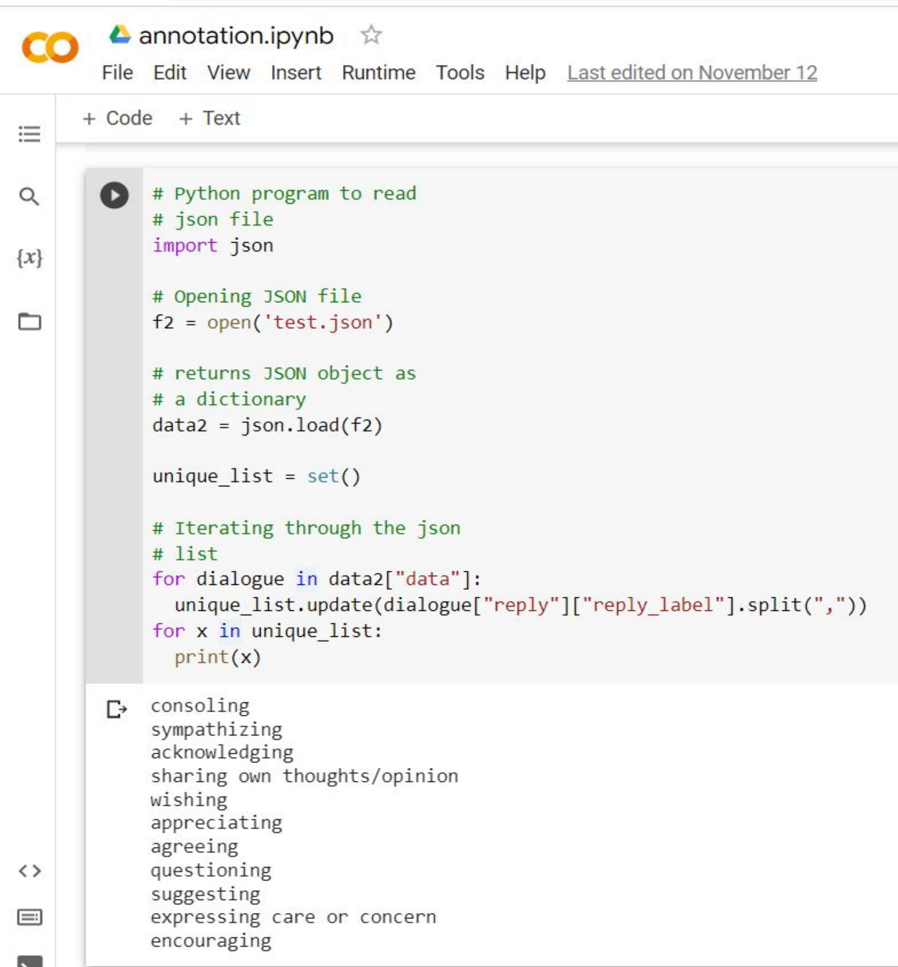
```
sh_demo.json > ...
{
    "data": [
        {
            "query": {
                "text": "Hey , you wanna see a movie tomorrow ?",
                "label": "happiness",
                "keywords": "see a movie tomorrow ?"
            },
            "reply": {
                "reply_label": "agreeing",
                "text": "Sounds like a good plan . What do you want to see ?"
            }
        },
        {
            "query": {
                "text": "How about Legally Blonde .",
                "label": "neutral",
                "keywords": "None"
            },
            "reply": {
                "reply_label": "sharing own thoughts/opinion",
                "text": "Ah , my girlfriend wanted to see that movie . I have to take her l
            }
        },
        {
            "query": {
                "text": "Isn't that a scary movie ?",
                "label": "neutral",
                "keywords": "None"
            },
            "reply": {
                "reply_label": "encouraging",
                "text": "How scary can it be ? Come on , it'll be fun ."
            }
        },
        {
            "query": {
                "text": "Ok . I'll give it a try ."
```

Python script:
https://colab.research.google.com/drive/10PZy4NSHa3CtM5VSa29tLSQ4OYQMldu5?usp=sharing

Annotating Emotion Classes in English Dataset (4,000+ records)



```
{
    "data": [
      {
        "query": {
          "text": "Hey , you wanna see a movie tomorrow ?",
          "label": "happiness",
          "keywords": "see a movie tomorrow ?"
        },
        "reply": {
          "reply_label": "agreeing",
          "text": "Sounds like a good plan . What do you want to see ?"
        }
      },
      {
        "query": {
          "text": "How about Legally Blonde .",
          "label": "neutral",
          "keywords": "None"
        },
        "reply": {
          "reply_label": "sharing own thoughts/opinion",
          "text": "Ah , my girlfriend wanted to see that movie . I have to t
        }
      },
      {
        "query": {
          "text": "Isn't that a scary movie ?",
          "label": "neutral",
          "keywords": "None"
        },
        "reply": {
          "reply_label": "encouraging",
          "text": "How scary can it be ? Come on , it'll be fun ."
        }
      },
      {
        "query": {
          "text": "Ok . I'll give it a try .",
```

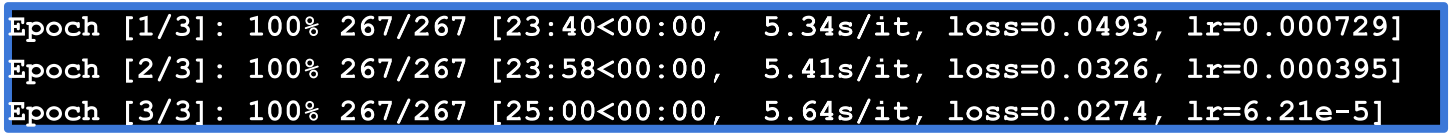


```python
# Python program to read
# json file
import json

# Opening JSON file
f2 = open('test.json')

# returns JSON object as
# a dictionary
data2 = json.load(f2)

unique_list = set()

# Iterating through the json
# list
for dialogue in data2["data"]:
  unique_list.update(dialogue["reply"]["reply_label"].split(","))
for x in unique_list:
  print(x)
```

```
consoling
sympathizing
acknowledging
sharing own thoughts/opinion
wishing
appreciating
agreeing
questioning
suggesting
expressing care or concern
encouraging
```

English Dataset Training Parameters & Results (BERT)

Language: English
Number of records: 4,269
Model: bert-base-uncased
Learning rate: 0.001
Number of epochs: 3
Training batch size: 16
Validation batch size: 16

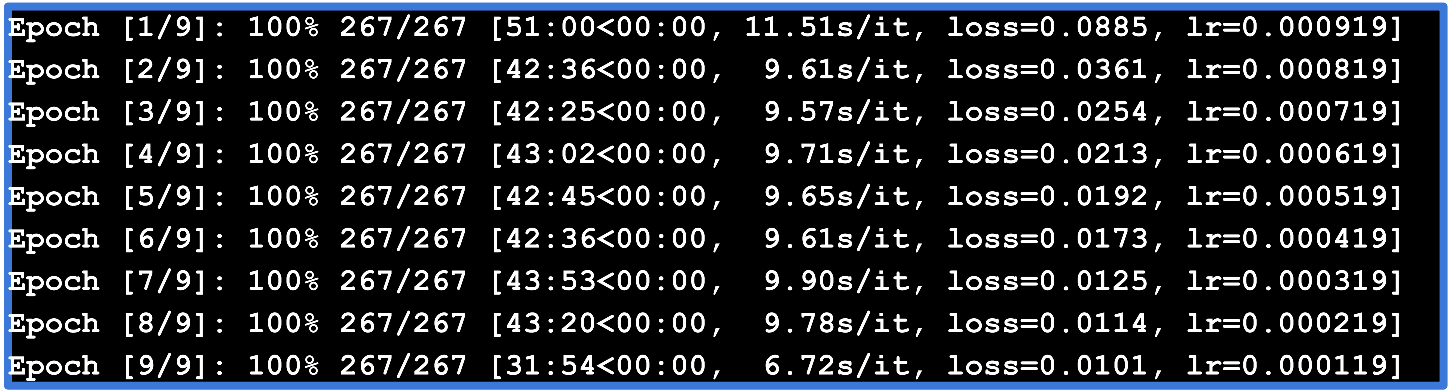


```
Epoch [1/3]: 100% 267/267 [23:40<00:00,  5.34s/it, loss=0.0493, lr=0.000729]
Epoch [2/3]: 100% 267/267 [23:58<00:00,  5.41s/it, loss=0.0326, lr=0.000395]
Epoch [3/3]: 100% 267/267 [25:00<00:00,  5.64s/it, loss=0.0274, lr=6.21e-5]
```

Reference: https://drive.google.com/file/d/1epM6283zJp70pNatrubRkP5iO3EmbdxT/view?usp=sharing

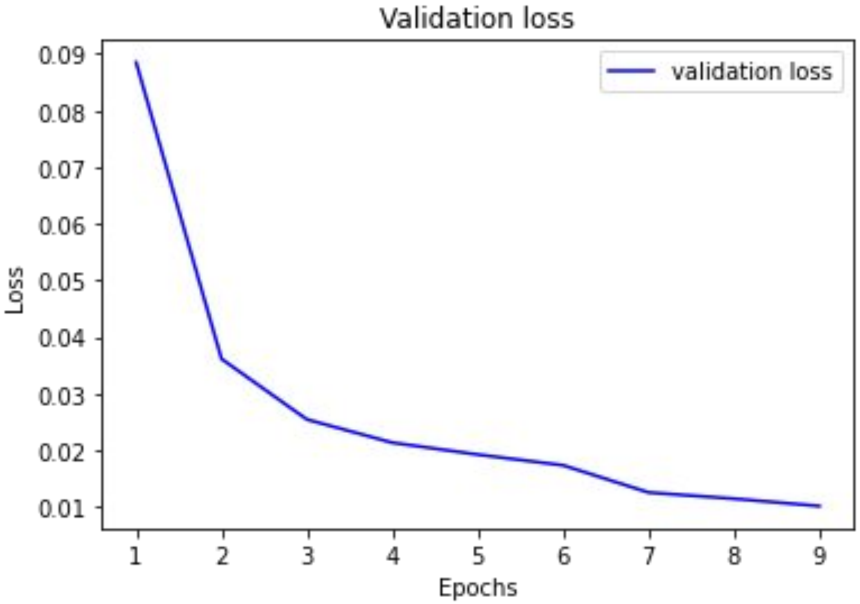
English Dataset Training Parameters & Results (GPT-2)

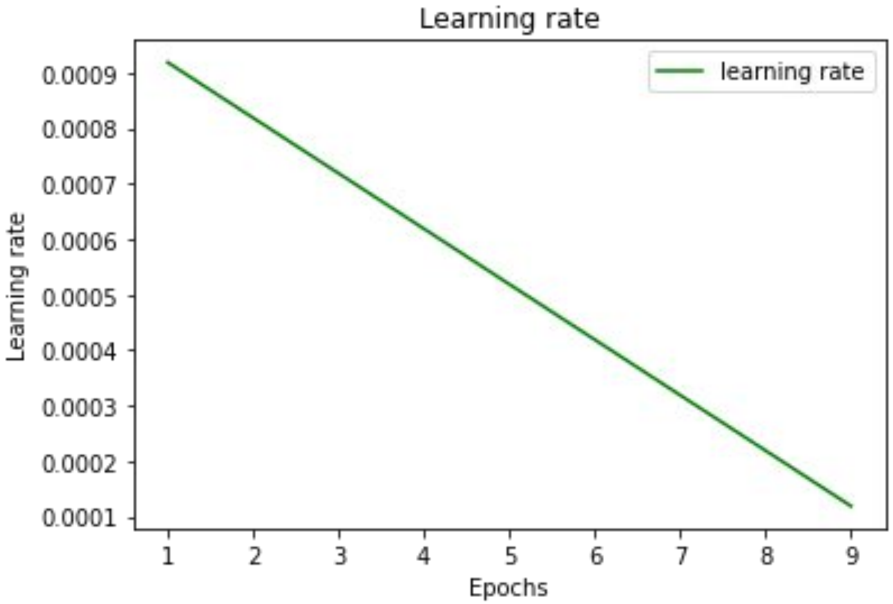
Language: English
Number of records: 4,269
Model: gpt2
Learning rate: 0.001
Number of epochs: 9
Training batch size: 16
Validation batch size: 16



```
Epoch [1/9]: 100% 267/267 [51:00<00:00, 11.51s/it, loss=0.0885, lr=0.000919]
Epoch [2/9]: 100% 267/267 [42:36<00:00,  9.61s/it, loss=0.0361, lr=0.000819]
Epoch [3/9]: 100% 267/267 [42:25<00:00,  9.57s/it, loss=0.0254, lr=0.000719]
Epoch [4/9]: 100% 267/267 [43:02<00:00,  9.71s/it, loss=0.0213, lr=0.000619]
Epoch [5/9]: 100% 267/267 [42:45<00:00,  9.65s/it, loss=0.0192, lr=0.000519]
Epoch [6/9]: 100% 267/267 [42:36<00:00,  9.61s/it, loss=0.0173, lr=0.000419]
Epoch [7/9]: 100% 267/267 [43:53<00:00,  9.90s/it, loss=0.0125, lr=0.000319]
Epoch [8/9]: 100% 267/267 [43:20<00:00,  9.78s/it, loss=0.0114, lr=0.000219]
Epoch [9/9]: 100% 267/267 [31:54<00:00,  6.72s/it, loss=0.0101, lr=0.000119]
```

English Dataset Results & Plots (GPT-2)


Validation loss


Learning rate

Future Scope

Experimenting with Stochastic Gradient Descent optimizer

Switching to GPU for training

Compare prediction results with novelty

Response generation & predictions

Minimize generalization error

References

- https://towardsdatascience.com/openai-gpt-2-understanding-language-generation-through-visualization-8252f683b2f8
- https://towardsdatascience.com/how-to-fine-tune-gpt-2-for-text-generation-ae2ea53bc272
- https://huggingface.co/bert-base-uncased
- https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForMaskedLM.forward
- https://github.com/XiaoMi/emma/tree/master/data/raw
- https://github.com/declare-lab/RECCON/blob/main/data/original_annotation/dailydialog_train.json
- https://towardsdatascience.com/cross-entropy-negative-log-likelihood-and-all-that-jazz-47a95bd2e81
- https://www.geeksforgeeks.org/intuition-of-adam-optimizer/
- https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c
- https://www.scss.tcd.ie/~koidlk/cs4062/Loss-Functions.pdf
- https://jalammar.github.io/illustrated-gpt2/

Thank You!..