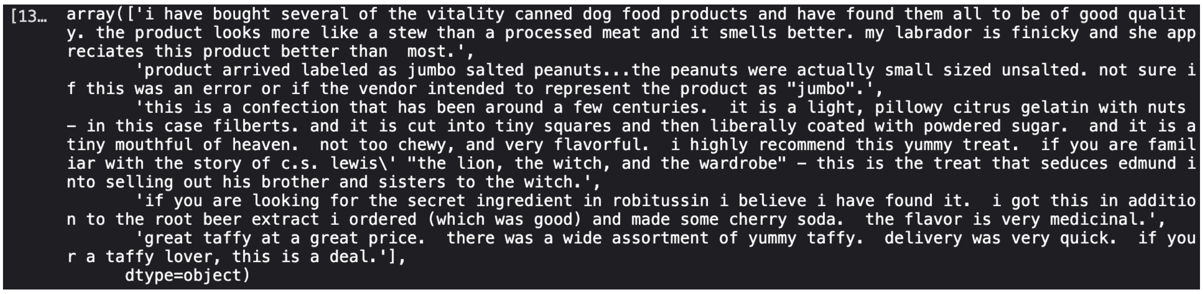
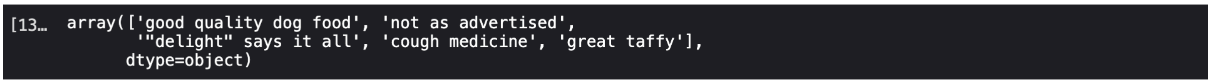
REPORT

1. Cleaning and preprocessing of the data was done i.e. all the NaN values and duplicates were removed. Thereafter, all the html tags were removed and all the text was then changed to lowercase.

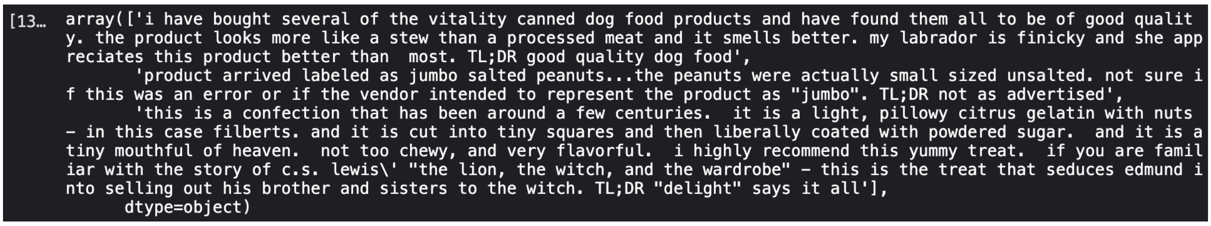


```
[13…  array(['i have bought several of the vitality canned dog food products and have found them all to be of good qualit
      y. the product looks more like a stew than a processed meat and it smells better. my labrador is finicky and she app
      reciates this product better than  most.',
             'product arrived labeled as jumbo salted peanuts...the peanuts were actually small sized unsalted. not sure i
      f this was an error or if the vendor intended to represent the product as "jumbo".',
             'this is a confection that has been around a few centuries.  it is a light, pillowy citrus gelatin with nuts
      - in this case filberts. and it is cut into tiny squares and then liberally coated with powdered sugar.  and it is a
      tiny mouthful of heaven.  not too chewy, and very flavorful.  i highly recommend this yummy treat.  if you are famil
      iar with the story of c.s. lewis\' "the lion, the witch, and the wardrobe" - this is the treat that seduces edmund i
      nto selling out his brother and sisters to the witch.',
             'if you are looking for the secret ingredient in robitussin i believe i have found it.  i got this in additio
      n to the root beer extract i ordered (which was good) and made some cherry soda.  the flavor is very medicinal.',
             'great taffy at a great price.  there was a wide assortment of yummy taffy.  delivery was very quick.  if you
      r a taffy lover, this is a deal.'],
            dtype=object)
```



```
[13…  array(['good quality dog food', 'not as advertised',
             '"delight" says it all', 'cough medicine', 'great taffy'],
            dtype=object)
```

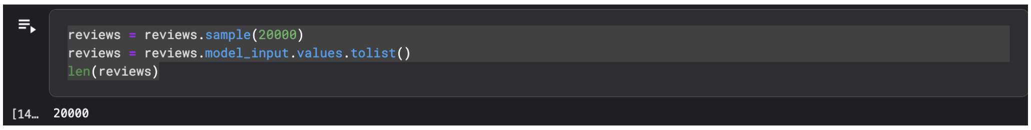
2. Then I made a column consisting of the Modified Input for the training i.e. concatenating ‘Text’ + ‘ TL;DR ‘ + ‘Summary’.



```
[13…  array(['i have bought several of the vitality canned dog food products and have found them all to be of good qualit
      y. the product looks more like a stew than a processed meat and it smells better. my labrador is finicky and she app
      reciates this product better than  most. TL;DR good quality dog food',
             'product arrived labeled as jumbo salted peanuts...the peanuts were actually small sized unsalted. not sure i
      f this was an error or if the vendor intended to represent the product as "jumbo". TL;DR not as advertised',
             'this is a confection that has been around a few centuries.  it is a light, pillowy citrus gelatin with nuts
      - in this case filberts. and it is cut into tiny squares and then liberally coated with powdered sugar.  and it is a
      tiny mouthful of heaven.  not too chewy, and very flavorful.  i highly recommend this yummy treat.  if you are famil
      iar with the story of c.s. lewis\' "the lion, the witch, and the wardrobe" - this is the treat that seduces edmund i
      nto selling out his brother and sisters to the witch. TL;DR "delight" says it all'],
            dtype=object)
```

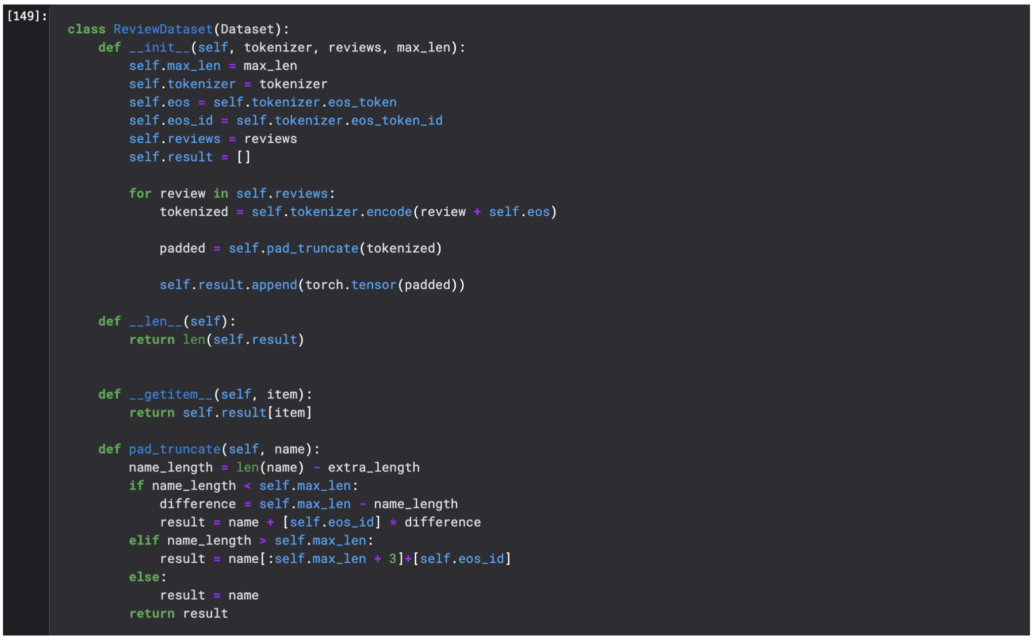
3. After that GPT2 model and tokenizer from hugging face were imported.

4. As the data was too much I took a subset of about 15000 entries and had a 75:25 ratio for training and testing the GPT2 model with a custom review dataset class.



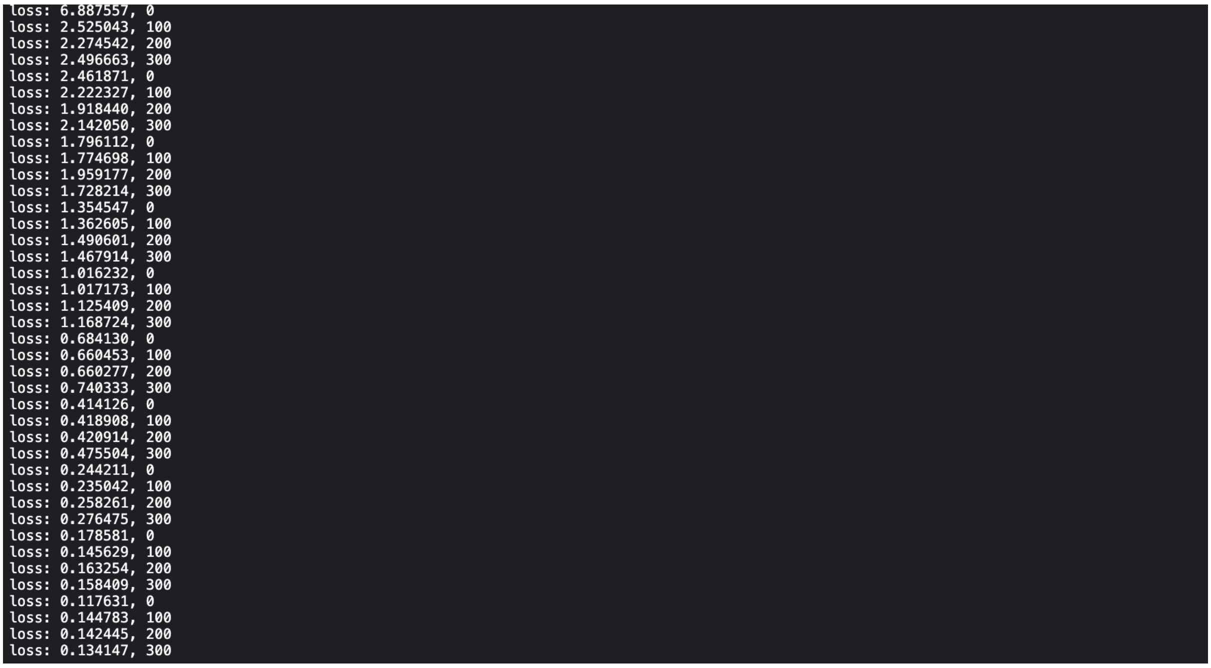
```
reviews = reviews.sample(20000)
reviews = reviews.model_input.values.tolist()
len(reviews)
```

```
[14…  20000
```



```
[149]:
class ReviewDataset(Dataset):
    def __init__(self, tokenizer, reviews, max_len):
        self.max_len = max_len
        self.tokenizer = tokenizer
        self.eos = self.tokenizer.eos_token
        self.eos_id = self.tokenizer.eos_token_id
        self.reviews = reviews
        self.result = []

        for review in self.reviews:
            tokenized = self.tokenizer.encode(review + self.eos)

            padded = self.pad_truncate(tokenized)

            self.result.append(torch.tensor(padded))

    def __len__(self):
        return len(self.result)


    def __getitem__(self, item):
        return self.result[item]

    def pad_truncate(self, name):
        name_length = len(name) - extra_length
        if name_length < self.max_len:
            difference = self.max_len - name_length
            result = name + [self.eos_id] * difference
        elif name_length > self.max_len:
            result = name[:self.max_len + 3]+[self.eos_id]
        else:
            result = name
        return result
```

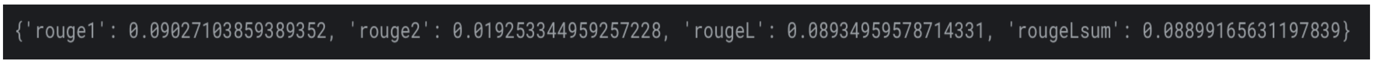
5. Which was then trained and fine tuned for better learning with a batch_size = 32 and with epoch = 10, the model was successfully learning as we can clearly see the loss being decreased significantly in 10 epochs.



```
loss: 6.887557, 0
loss: 2.525043, 100
loss: 2.274542, 200
loss: 2.496663, 300
loss: 2.461871, 0
loss: 2.222327, 100
loss: 1.918440, 200
loss: 2.142050, 300
loss: 1.796112, 0
loss: 1.774698, 100
loss: 1.959177, 200
loss: 1.728214, 300
loss: 1.354547, 0
loss: 1.362605, 100
loss: 1.490601, 200
loss: 1.467914, 300
loss: 1.016232, 0
loss: 1.017173, 100
loss: 1.125409, 200
loss: 1.168724, 300
loss: 0.684130, 0
loss: 0.660453, 100
loss: 0.660277, 200
loss: 0.740333, 300
loss: 0.414126, 0
loss: 0.418908, 100
loss: 0.420914, 200
loss: 0.475504, 300
loss: 0.244211, 0
loss: 0.235042, 100
loss: 0.258261, 200
loss: 0.276475, 300
loss: 0.178581, 0
loss: 0.145629, 100
loss: 0.163254, 200
loss: 0.158409, 300
loss: 0.117631, 0
loss: 0.144783, 100
loss: 0.142445, 200
loss: 0.134147, 300
```

6. After that the model was saved as a pickle file.

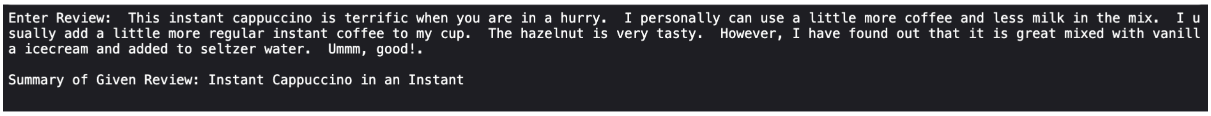
7. Then functions for calculating the top k choices and then for generating the summary calculates summaries for the test set and rouge scores are then calculated.



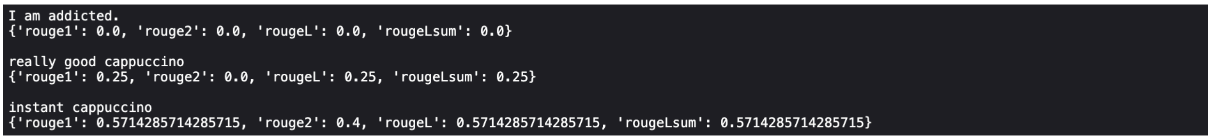
```
{'rouge1': 0.09027103859389352, 'rouge2': 0.019253344959257228, 'rougeL': 0.08934959578714331, 'rougeLsum': 0.08899165631197839}
```

8. **Observation:** Rouge Scores are not the best but the summaries are pretty relevant to the texts given and have significantly improved as the optimization of the model was done.

9. Lastly any review from the test set can be inputted then 3 relevant summaries of the review are generated and a combined and individual rouge score is calculated.



```
Enter Review:  This instant cappuccino is terrific when you are in a hurry.  I personally can use a little more coffee and less milk in the mix.  I u
sually add a little more regular instant coffee to my cup.  The hazelnut is very tasty.  However, I have found out that it is great mixed with vanill
a icecream and added to seltzer water.  Ummm, good!.

Summary of Given Review: Instant Cappuccino in an Instant
```



```
I am addicted.
{'rouge1': 0.0, 'rouge2': 0.0, 'rougeL': 0.0, 'rougeLsum': 0.0}

really good cappuccino
{'rouge1': 0.25, 'rouge2': 0.0, 'rougeL': 0.25, 'rougeLsum': 0.25}

instant cappuccino
{'rouge1': 0.5714285714285715, 'rouge2': 0.4, 'rougeL': 0.5714285714285715, 'rougeLsum': 0.5714285714285715}
```

References:

I took some help from the links mentioned below:

1. https://medium.com/@eren9677/text-summarization-387836c9e178
2. https://towardsdatascience.com/text-summarization-with-gpt2-and-layer-ai-599625085d8e