Predicting the character from The Simpsons

Ruchina Shakya
Yingfan Wang

Outline

- What we have tried and why they didn’t work
- Why BERT?
- Steps
- Results and Problems

What we have tried

- Word2vec
  - We got the most similar words of each character, but that’s not enough to predict a whole sentence
- Sentiment analysis
  - Sentiment analysis could only be divided into positive,negative and neutral sentiment whereas we were looking into six different characters.
  - The character could have same ratio of sentiments.

Why BERT?

- We came across coronavirus tweets NLP which used BERT for the text classification(resources).
- Pretrained model
- Process large amount of text
- Keras API handles the padding and masking of the data(Dense layer).

Steps:

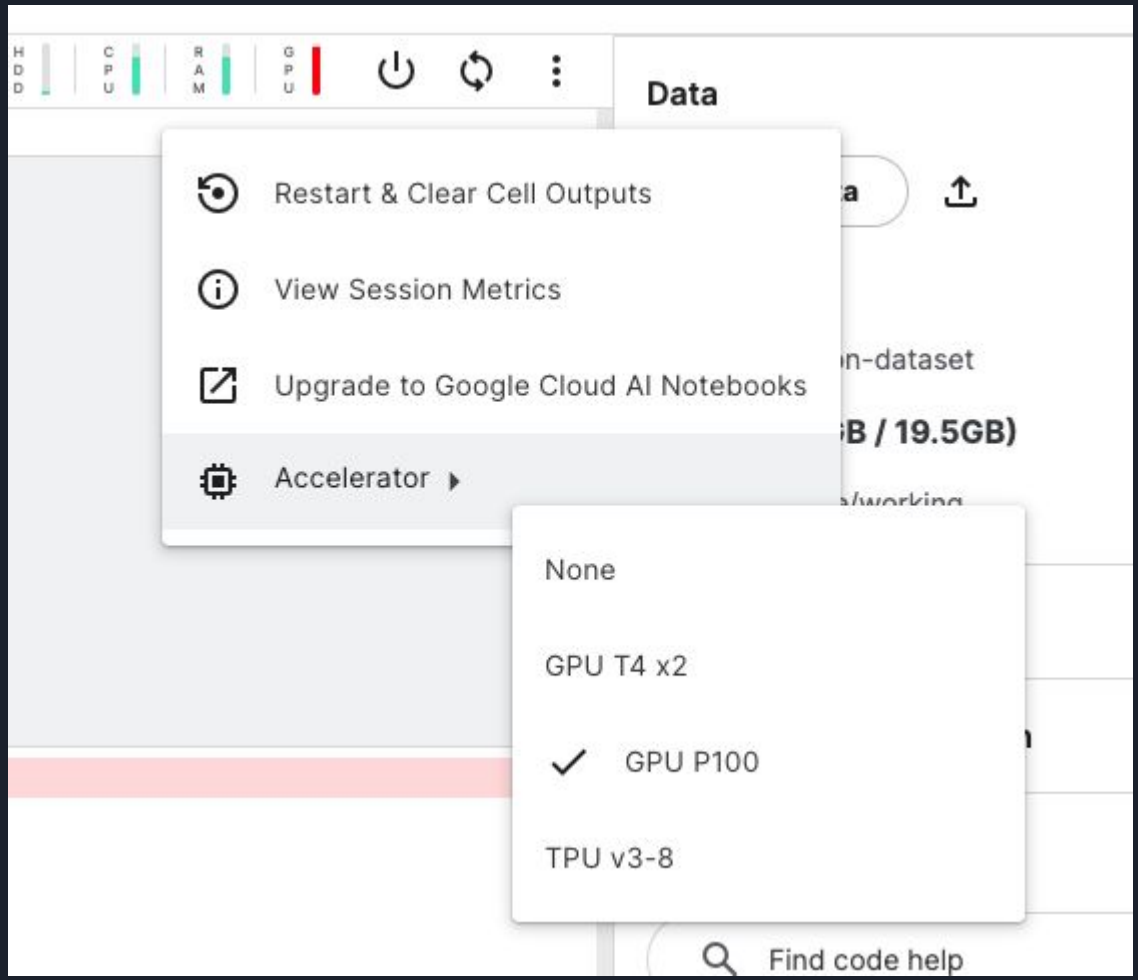
- Importing dataset
- Cleaning data(removing missing value,non-alphabetical characters,stopwords)
- Labeling class,encoding text, building vocab file
- creating a bert embedding layer
- Set parameters and build a model
- Running
- Exporting output as a csv file

Results:

- The total number of submission was 3.
- Used different number of epoch(3,5,10) and cleaned the dataset to improve the accuracy.
- The best accuracy was 0.50127.

Problems:

- Overfitting. Increasing the epoch increased the accuracy but the validation accuracy was not improved.
- On a local machine with epoch:3 , it took 21 hours.
- The validation accuracy fluctuates.


HDD
CPU
RAM
GPU
Data
Restart & Clear Cell Outputs
View Session Metrics
Upgrade to Google Cloud AI Notebooks
Accelerator
None
GPU T4 x2
GPU P100
TPU v3-8
Find code help

Kaggle notebook with GPU P100

```
Epoch 1/3
1217/1217 [==============================] - 1271s 1s/step - loss: 1.4211 - accuracy: 0.4016 - val_loss: 1.2735 - val_accuracy: 0.4733

Epoch 00001: val_accuracy improved from -inf to 0.47333, saving model to model.h5
Epoch 2/3
1217/1217 [==============================] - 1258s 1s/step - loss: 1.2704 - accuracy: 0.4846 - val_loss: 1.2515 - val_accuracy: 0.4957

Epoch 00002: val_accuracy improved from 0.47333 to 0.49573, saving model to model.h5
Epoch 3/3
 319/1217 [======>.......................] - ETA: 14:16 - loss: 1.1084 - accuracy: 0.5644
```

Jupyter Notebook

```
Epoch 1/5
2022-10-29 10:01:56.955037: W tensorflow/core/platform/profile_utils/cpu_utils.cc:128] Failed to get CPU frequency:
0 Hz
1217/1217 [==============================] - ETA: 0s - loss: 1.4119 - accuracy: 0.4043
Epoch 1: val_accuracy improved from -inf to 0.47466, saving model to model.h5
1217/1217 [==============================] - 26522s 22s/step - loss: 1.4119 - accuracy: 0.4043 - val_loss: 1.2737 -
val_accuracy: 0.4747
Epoch 2/5
1217/1217 [==============================] - ETA: 0s - loss: 1.2612 - accuracy: 0.4848
Epoch 2: val_accuracy improved from 0.47466 to 0.49214, saving model to model.h5
1217/1217 [==============================] - 26390s 22s/step - loss: 1.2612 - accuracy: 0.4848 - val_loss: 1.2338 -
val_accuracy: 0.4921
Epoch 3/5
1217/1217 [==============================] - ETA: 0s - loss: 1.1255 - accuracy: 0.5611
Epoch 3: val_accuracy improved from 0.49214 to 0.49450, saving model to model.h5
1217/1217 [==============================] - 26386s 22s/step - loss: 1.1255 - accuracy: 0.5611 - val_loss: 1.2735 -
```

References:

https://www.kaggle.com/code/nayansakhiya/text-classification-using-bert/notebook

https://www.kaggle.com/code/phongphamds/word2vec-using-gensim-library

Thank You!