sprint3_review_generator

December 23, 2022

1 Review generator

[1]:
```
import pandas as pd
import seaborn as sn
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.transforms import Bbox
from random import shuffle
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow import keras
```

First we read in the reviews

[46]:
```
reviews=pd.read_csv("../tripadvisor_dataset/reviews.csv")
reviews.shape
```

[46]: (136173, 6)

We're going to take a small part of the data to train, we've first tried to train it on all data but after a few hours we couldn't even get 1 epoch done. We decided to make it smaller.

We split the positive and negatives so we can have an equal distribution. We think that if we do that we'll be able to generate positive and negative reviews.

[3]:
```
reviews_positive = reviews[reviews["rating"] == 5]
reviews_negative = reviews[reviews["rating"] == 1]
print(reviews_negative.shape)
reviews_positive.shape
```

```
(8060, 6)
```

[3]: (58117, 6)

When we look at the positive and negative reviews, we see there is a mix of languages. This is not the best way to train this model. Sadly we found this out after training the model, so this model is trained on multiple languages. In the new version we'll only train on English.

[25]:
```
reviews_positive.head(5)
```

[25]:
```
         id      reviewer name  \
0  13969825           bertd818
1  13969825       593laetitiad
2  13969825           612ellen
3  13969825             j0ycal
4  13969825  Global45882037169

                                                title                date  \
0                      supper snelle en lekkere lunch  September 26, 2022
1     Un délicieux repas aux saveurs de la Thaïlande  September 24, 2022
2                       Altijd leuk om terug te komen  September 19, 2022
3  Perfect onthaal/gastvrijheid, superlekker eten…  September 19, 2022
4                                          genieten  September 19, 2022

   rating                                             review
0     5.0  supper lekker gegeten tijdens de middag, als w…
1     5.0  Un menu lunch très bien équilibré aux niveaux …
2     5.0  Super gezellig restaurant met super bediening …
3     5.0  Perfect onthaal, lekker eten. Heel goede lunch…
4     5.0  verrassend lekker gegeten, een mooi en rustig …
```

[66]:
```
reviews_negative.head(5)
```

[66]:
```
            id   reviewer name                                  title  \
57   13969825     398maartjeg  Wel ok, maar zeker niet de allerbeste
118  13969825  michellD9555DF                    Waardeloze service
141  13969825      143andreic                             Ungenügend
208  13969825          559zul                        teleurstelling
494  13969825      sv20172017           Rendement per m2 optimaal
637  13969825      pierrecvdv                             lamentable
795  13969825         Rajan N               Uncomfortable seating
809  13969825         ulibear          super unfriendly headwaiter
926  13969825  annerasschaert                             Jammer…
938  13969825  annerasschaert                             Jammer…

                    date  rating  \
57         June 11, 2022     1.0
118        March 5, 2022     1.0
141      January 6, 2022     1.0
208    September 9, 2021     1.0
494      January 3, 2020     1.0
637        July 14, 2019     1.0
795  September 24, 2018     1.0
809  September 17, 2018     1.0
926       April 16, 2018     1.0
```

```
938      April 16, 2018     1.0

                                                  review
57   Bediening is aardig maar eten is niet heel spe…
118  Niet eens aan eten toegekomen. Je reserveert e…
141  Wir haben nicht reserviert kamen rein und habe…
208  Be ready to wait LONG for your food.\nWe had t…
494  Na het lezen van de recentie van Gault & Milla…
637  als u slecht wilt eten, voor veel geld, is kin…
795  After eating in few restaurant in the dark str…
809  We have reserved a table for 4. We then inform…
926  Gisteren vol verwachtingen naar Kin Khao… he…
938  Gisteren vol verwachtingen naar Kin Khao… he…
```

The number 2500 is choosen after trial and error to see how long one epoch would take. With 5000 total reviews, 1 epoch takes around 30mins which is acceptable.

[4]:
```
reviews_positive = reviews_positive.head(2500)
reviews_negative = reviews_negative.head(2500)
reviews = pd.concat([reviews_negative, reviews_positive])
reviews
```

[4]:
```
            id   reviewer name                                    title  \
57    13969825     398maartjeg  Wel ok, maar zeker niet de allerbeste
118   13969825  michellD9555DF                     Waardeloze service
141   13969825      143andreic                             Ungenügend
208   13969825          559zul                          teleurstelling
494   13969825      sv20172017              Rendement per m2 optimaal
…        …             …                                        …
2946  10032417       593sannew                          Excellent !
2947  10032417      513veerlev                   Love the experience
2948  10032417       76sandrar                Lovely Lunch and Owner!
2949  10032417        PuchPuch      Tres bon goûter sur une agréable
2952  10032417        Galitea7                             Magnifique

                   date  rating  \
57        June 11, 2022     1.0
118       March 5, 2022     1.0
141     January 6, 2022     1.0
208   September 9, 2021     1.0
494     January 3, 2020     1.0
…                  …      …
2946     August 6, 2018     5.0
2947     August 3, 2018     5.0
2948      July 30, 2018     5.0
2949       July 7, 2018     5.0
2952      June 24, 2018     5.0
```

```
                                                  review
57    Bediening is aardig maar eten is niet heel spe…
118   Niet eens aan eten toegekomen. Je reserveert e…
141   Wir haben nicht reserviert kamen rein und habe…
208   Be ready to wait LONG for your food.\nWe had t…
494   Na het lezen van de recentie van Gault & Milla…
…                                                   …
2946  Wat een prachtige plek en een enorm toffe eige…
2947  It is not a place where you just have somethin…
2948  Not only is the place super cute and original,…
2949  Le lieu mérite une visite ne serait-ce que pou…
2952  Maison Elza… Entdeckt vom Wasser aus…di…

[5000 rows x 6 columns]
```

Writing the reviews to a txt file to access it later

[5]:
```
reviews_txt = open("reviews.txt", "w", encoding="utf-8")

reviews_cleaned = ""
for rev in list(reviews["review"]):
    rev = str(rev)
    reviews_txt.write(rev + "\n")

reviews_txt.close()
with open("reviews.txt", "r", encoding="utf-8") as f:
    reviews_cleaned = f.read()
```

[6]:
```
print(len(reviews_cleaned))
```

```
2159953
```

1.1 Preprocessing data

[7]:
```
tokenizer = keras.preprocessing.text.Tokenizer(char_level=True)
tokenizer.fit_on_texts(reviews_cleaned)
max_id = len(tokenizer.word_index)
```

[8]:
```
dataset_size = tokenizer.document_count # total number of characters
[encoded] = np.array(tokenizer.texts_to_sequences([reviews_cleaned])) - 1
train_size = dataset_size * 90 // 100
dataset = tf.data.Dataset.from_tensor_slices(encoded[:train_size])
```

[9]:
```
n_steps = 100
window_length = n_steps + 1 # target = input shifted 1 character ahead
dataset = dataset.window(window_length, shift=1, drop_remainder=True)
```

[10]:
```
dataset = dataset.flat_map(lambda window: window.batch(window_length))

batch_size = 32
dataset = dataset.shuffle(10000).batch(batch_size)
dataset = dataset.map(lambda windows: (windows[:, :-1], windows[:, 1:]))
```

[11]:
```
dataset = dataset.map(
    lambda X_batch, Y_batch: (tf.one_hot(X_batch, depth=max_id), Y_batch))
```

[12]:
```
dataset = dataset.prefetch(1)
for X_batch, Y_batch in dataset.take(1):
    print(X_batch.shape, Y_batch.shape)
```

```
(32, 100, 830) (32, 100)
```

1.2 Training the model

[13]:
```
model = keras.models.Sequential([
    keras.layers.GRU(128, return_sequences=True, input_shape=[None, max_id],
                     #dropout=0.2, recurrent_dropout=0.2),
                     dropout=0.2),
    keras.layers.GRU(128, return_sequences=True,
                     #dropout=0.2, recurrent_dropout=0.2),
                     dropout=0.2),
    keras.layers.TimeDistributed(keras.layers.Dense(max_id,
                                                    activation="softmax"))
])
model.compile(loss="sparse_categorical_crossentropy", optimizer="adam")
history = model.fit(dataset, epochs=10)
```

```
Epoch 1/10

2022-12-19 16:39:36.275348: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384]
Loaded cuDNN version 8500

60746/60746 [==============================] - 1933s 32ms/step - loss: 1.7735
Epoch 2/10
60746/60746 [==============================] - 1900s 31ms/step - loss: 1.6156
Epoch 3/10
60746/60746 [==============================] - 2150s 35ms/step - loss: 1.5940
Epoch 4/10
60746/60746 [==============================] - 1917s 32ms/step - loss: 1.5833
Epoch 5/10
60746/60746 [==============================] - 1946s 32ms/step - loss: 1.5765
Epoch 6/10
60746/60746 [==============================] - 1951s 32ms/step - loss: 1.5725
Epoch 7/10
60746/60746 [==============================] - 2207s 36ms/step - loss: 1.5676
```

```
Epoch 8/10
60746/60746 [==============================] - 1978s 33ms/step - loss: 1.5648
Epoch 9/10
60746/60746 [==============================] - 1974s 32ms/step - loss: 1.5621
Epoch 10/10
60746/60746 [==============================] - 1968s 32ms/step - loss: 1.5603
```

1.3 Save the model

[ ]:
```
model.save('./review_generator_v2/')
```

Reading in the model that was trained above

[14]:
```
model = keras.models.load_model("./review_generator_v2")
```

[15]:
```
def preprocess(texts):
    X = np.array(tokenizer.texts_to_sequences(texts)) - 1
    return tf.one_hot(X, max_id)
```

Trying a simple example

[52]:
```
X_new = preprocess(["How are yo"])
#Y_pred = model.predict_classes(X_new)
Y_pred = np.argmax(model(X_new), axis=-1)
tokenizer.sequences_to_texts(Y_pred + 1)[0][-1] # 1st sentence, last char
```

[52]:
```
'u'
```

[53]:
```
def next_char(text, temperature=1):
    X_new = preprocess([text])
    y_proba = model(X_new)[0, -1:, :]
    rescaled_logits = tf.math.log(y_proba) / temperature
    char_id = tf.random.categorical(rescaled_logits, num_samples=1) + 1
    return tokenizer.sequences_to_texts(char_id.numpy())[0]
```

[54]:
```
def complete_text(text, n_chars=50, temperature=1):
    for _ in range(n_chars):
        text += next_char(text, temperature)
    return text
```

1.4 Temperature

The temperature is an import parameter that determines.

A high temperature forces the model to make more original predictions. A low temperature make sures that the model doesn't go off topic, mostly the same text is being predicted if the temperature is very low.

Now we test what our review generator can do

[58]:
```
print(complete_text("lekker", 50, temperature=0.2))
```

```
lekkere gerechten en de koks zelf geserveerd door de kok
```

[34]:
```
print(complete_text("aanrader", 50, temperature=0.2))
```

```
aanrader! het was een zeer geslaagd op de keuken van de ko
```

That looks decent! But it can also generate nonsense, especially when we put a high temperature

[62]:
```
print(complete_text("super lekker", 50, temperature=1))
```

```
super lekkere witneervaste wat een ongedwong  richtigd met een
```

And the reviews aren't always logical

[19]:
```
print(complete_text("good food", 100, temperature=0.2))
```

```
good food. the parth share was a plate of the evening and the perfectly prepared
with an excellent food and t
```

[20]:
```
print(complete_text("The food was awful!", 100, temperature=0.2))
```

```
The food was awful! the service was a so good and cream of the past of the
patershol was also food was a pleasure and t
```

We also weren't really be able to generate negative reviews, even though it was half our dataset.

[77]:
```
print(complete_text("trage bediening", 50, temperature=0.2))
```

```
trage bediening. de kok is een aangepaste wijnen. de kok is een z
```

[78]:
```
print(complete_text("slecht eten", 50, temperature=0.2))
```

```
slecht eten en de koks werd ook geen steeds een aangepaste wi
```

1.5 Conclusion

This model didn't perform well when generating negative reviews. Generating positive reviews are better, but not optimal. A reason for this might be because we mixed multiple languages. Another reason is that we train this model from 0, and it probably needs way more data. We couldn't train much data because the training time is high.

We could solve this issue by using a pretrained model that can already understand English. Then we can finetune it to a specific task (generate reviews). We just finetune an existing model with our dataset. The model we finetuned is GPT-2.

In this file `review_generator_v3.ipynb` you can find the sequel.

1.6 References:

https://github.com/ageron/handson-ml2