lab5

October 10, 2023

[76]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
import torch
from sklearn.model_selection import train_test_split
from torch.utils.data import Dataset, DataLoader
from transformers import AdamW, BertTokenizer, BertForSequenceClassification
from transformers import DistilBertTokenizer,␣
 ↪DistilBertForSequenceClassification, AdamW, get_linear_schedule_with_warmup
from tqdm import tqdm
from torch.nn.utils.rnn import pad_sequence
import torch
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

[77]:
```
pd.set_option('display.max_colwidth', 2000)
pd.set_option('display.max_rows', 2000)
# Load the data
file_path = './dataSets/IMDB Dataset.csv'
data = pd.read_csv(file_path)

# Display the first row to confirm the data is loaded correctly
display(data.head())

# This will hold the new structured data
new_data = {
    'review': [],
    'sentiment': []
}

# Iterate through each row in the data
for index, row in data.iterrows():
    # Split the review text into individual reviews based on <br /><br />
    reviews = row[0].split('<br /><br />')
    # Extend the new_data with these reviews and the associated sentiment
    new_data['review'].extend(reviews)
    # Repeat the sentiment for each review
```

```
    new_data['sentiment'].extend([row[1]] * len(reviews))

# Create a new DataFrame
new_df = pd.DataFrame(new_data)

# Display the first 5 rows to confirm the data is loaded correctly
display(new_df.head())

def preprocess_text(text):
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.lower()
    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(text)
    filtered_text = [word for word in word_tokens if word not in stop_words]
    filtered_text = [word for word in filtered_text if len(word) >= 3]
    text = " ".join(filtered_text)
    return text

small_df = new_df.sample(n=3000, random_state=42)

# Apply preprocessing and reduce sequence length
small_df['review'] = small_df['review'].apply(preprocess_text)
small_df['review'] = small_df['review'].apply(lambda x: x[:512])

# Convert sentiments to binary labels
small_df['sentiment'] = small_df['sentiment'].apply(lambda x: 1 if x ==␣
 ↪'positive' else 0)

# Split the data
train_texts, test_texts, train_labels, test_labels = train_test_split(
    small_df['review'].tolist(),
    small_df['sentiment'].tolist(),
    test_size=0.2,  # 80% training, 20% testing
    random_state=42
)
```

```
                   review  \
0  One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me.<br /><br />The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. This show pulls no punches with regards to drugs, sex or violence. Its is hardcore, in the classic use of the word.<br /><br />It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentary. It focuses mainly on Emerald City, an experimental section of the prison where all the cells have glass fronts and face inwards, so privacy is not high on the agenda. Em City is home to many..Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and more…so scuffles, death stares, dodgy dealings and shady agreements are never far away.<br /><br />I would say the main appeal of the show is due to the fact that it goes where other shows wouldn't dare. Forget pretty pictures painted for mainstream audiences, forget charm, forget romance…OZ doesn't mess around. The first episode I ever saw struck me as so nasty it was surreal, I couldn't say I was ready for it, but as I watched more, I developed a taste for Oz, and got accustomed to the high levels of graphic violence. Not just violence, but injustice (crooked guards who'll be sold out for a nickel, inmates who'll kill on order and get away with it, well mannered, middle class inmates being turned into prison bitches due to their lack of street skills or prison experience) Watching Oz, you may become comfortable with what is uncomfortable viewing…thats if you can get in touch with your darker side.
```

1

A wonderful little production.

The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece.

The actors are extremely well chosen- Michael Sheen not only "has got all the polari" but he has all the voices down pat too! You can truly see the seamless editing guided by the references to Williams' diary entries, not only is it well worth the watching but it is a terrificly written and performed piece. A masterful production about one of the great master's of comedy and his life.

The realism really comes home with the little things: the fantasy of the guard which, rather than use the traditional 'dream' techniques remains solid then disappears. It plays on our knowledge and our senses, particularly with the scenes concerning Orton and Halliwell and the sets (particularly of their flat with Halliwell's murals decorating every surface) are terribly well done.

2

I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air conditioned theater and watching a light-hearted comedy. The plot is simplistic, but the dialogue is witty and the characters are likable (even the well bread suspected serial killer). While some may be disappointed when they realize this is not Match Point 2: Risk Addiction, I thought it was proof that Woody Allen is still fully in control of the style many of us have grown to love.

This was the most I'd laughed at one of Woody's comedies in years (dare I say a decade?). While I've never been impressed with Scarlet Johanson, in this she managed to tone down her "sexy" image and jumped right into a average, but spirited young woman.

This may not be the crown jewel of his career, but it was wittier than "Devil Wears Prada" and more interesting than "Superman" a great comedy to go see with friends.

3

Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his parents are fighting all the time.

This movie is slower than a soap opera… and suddenly, Jake decides to become Rambo and kill the zombie.

OK, first of all when you're going to make a film you must Decide if its a thriller or a drama! As a drama the movie is watchable. Parents are divorcing & arguing like in real life. And then we have Jake with his closet which totally ruins all the film! I expected to see a BOOGEYMAN similar movie, and instead i watched a drama with some meaningless thriller spots.

3 out of 10 just for the well playing parents & descent dialogs. As for the shots with Jake: just ignore them.

4

Petter Mattei's "Love in the Time of Money" is a visually stunning film to watch. Mr. Mattei offers us a vivid portrait about human relations. This is a movie that seems to be telling us what money, power and success do to people in the different situations we encounter.

This being a variation on the Arthur Schnitzler's play about the same theme, the director transfers the action to the present time New York where all these different characters meet and connect. Each one is connected in one way, or another to the next person, but no one seems to know the previous point of contact. Stylishly, the film has a sophisticated luxurious look. We are taken to see how these people live and the world they live in their own habitat.

The only thing one gets out of all these souls in the picture is the different stages of loneliness each one inhabits. A big city is not exactly the best place in which human relations find sincere fulfillment, as one discerns is the case with most of the people we encounter.

The acting is good under Mr. Mattei's direction. Steve Buscemi, Rosario Dawson, Carol Kane, Michael Imperioli, Adrian Grenier, and the rest of the talented cast, make these characters come alive.

We wish Mr. Mattei good luck and await anxiously for his next work.

```
  sentiment
0  positive
1  positive
2  positive
3  negative
4  positive
```

```
                                                                                ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                        review  \
0                                                                               ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                    One of the other reviewers has mentioned that␣
 ↪after watching just 1 Oz episode you'll be hooked. They are right, as this is␣
 ↪exactly what happened with me.
1                                                                               ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                                    The first thing that struck me about Oz␣
 ↪was its brutality and unflinching scenes of violence, which set in right from␣
 ↪the word GO. Trust me, this is not a show for the faint hearted or timid. This␣
 ↪show pulls no punches with regards to drugs, sex or violence. Its is hardcore,␣
 ↪in the classic use of the word.
```

```
2                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                      It is called OZ as␣
 ↪that is the nickname given to the Oswald Maximum Security State Penitentary.␣
 ↪It focuses mainly on Emerald City, an experimental section of the prison where␣
 ↪all the cells have glass fronts and face inwards, so privacy is not high on␣
 ↪the agenda. Em City is home to many..Aryans, Muslims, gangstas, Latinos,␣
 ↪Christians, Italians, Irish and more….so scuffles, death stares, dodgy dealings␣
 ↪and shady agreements are never far away.
3  I would say the main appeal of the show is due to the fact that it goes where␣
 ↪other shows wouldn't dare. Forget pretty pictures painted for mainstream␣
 ↪audiences, forget charm, forget romance…OZ doesn't mess around. The first␣
 ↪episode I ever saw struck me as so nasty it was surreal, I couldn't say I was␣
 ↪ready for it, but as I watched more, I developed a taste for Oz, and got␣
 ↪accustomed to the high levels of graphic violence. Not just violence, but␣
 ↪injustice (crooked guards who'll be sold out for a nickel, inmates who'll kill␣
 ↪on order and get away with it, well mannered, middle class inmates being␣
 ↪turned into prison bitches due to their lack of street skills or prison␣
 ↪experience) Watching Oz, you may become comfortable with what is uncomfortable␣
 ↪viewing….thats if you can get in touch with your darker side.
4                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                         ␣
 ↪                                                                        A␣
 ↪wonderful little production.

  sentiment
0  positive
1  positive
2  positive
3  positive
4  positive
```

[78]:

```
class MovieReviewDataset(Dataset):
    def __init__(self, texts, labels):
        self.texts = texts
        self.labels = labels

    def __len__(self):
```

```
        return len(self.texts)

    def __getitem__(self, idx):
        text = self.texts[idx]
        label = self.labels[idx]
        inputs = tokenizer(text, return_tensors='pt', truncation=True,␣
 ↪add_special_tokens=True)
        inputs['labels'] = torch.tensor(label)
        return inputs

# Collate function
def collate_fn(batch):
    input_ids = [item['input_ids'].squeeze(0) for item in batch]
    attention_masks = [item['attention_mask'].squeeze(0) for item in batch]
    labels = torch.tensor([item['labels'] for item in batch])
    input_ids = pad_sequence(input_ids, batch_first=True,␣
 ↪padding_value=tokenizer.pad_token_id)
    attention_masks = pad_sequence(attention_masks, batch_first=True,␣
 ↪padding_value=0)
    return {
        'input_ids': input_ids,
        'attention_mask': attention_masks,
        'labels': labels
    }

# Load tokenizer and model
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased',␣
 ↪do_lower_case=True)
model = BertForSequenceClassification.from_pretrained("bert-base-uncased",␣
 ↪num_labels=2)

# Move model to GPU if available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

# Create DataLoaders
train_dataset = MovieReviewDataset(train_texts, train_labels)
val_dataset = MovieReviewDataset(val_texts, val_labels)
train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True,␣
 ↪collate_fn=collate_fn)
val_loader = DataLoader(val_dataset, batch_size=16, collate_fn=collate_fn)

# Specify optimizer and scheduler
optimizer = AdamW(model.parameters(), lr=1e-5)
num_training_steps = len(train_loader) * 3
```

```
scheduler = get_linear_schedule_with_warmup(optimizer, num_warmup_steps=0,␣
 ↪num_training_steps=num_training_steps)

# Train the model
num_epochs = 3
for epoch in range(num_epochs):
    model.train()
    for batch in tqdm(train_loader):
        optimizer.zero_grad()
        inputs = {k: v.to(device) for k, v in batch.items() if k != 'labels'}
        labels = batch['labels'].to(device)
        outputs = model(**inputs, labels=labels)
        loss = outputs.loss
        loss.backward()
        optimizer.step()
        scheduler.step()
```

```
Some weights of BertForSequenceClassification were not initialized from the
model checkpoint at bert-base-uncased and are newly initialized:
['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
/Users/hudsonshimanyula/anaconda3/envs/AI_MASTERS_ENV/lib/python3.11/site-
packages/transformers/optimization.py:411: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch
implementation torch.optim.AdamW instead, or set `no_deprecation_warning=True`
to disable this warning
  warnings.warn(
100%|      | 150/150 [03:43<00:00,  1.49s/it]
100%|      | 150/150 [03:31<00:00,  1.41s/it]
100%|      | 150/150 [03:31<00:00,  1.41s/it]
```

[79]:
```
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
# Evaluate the model on the validation set
model.eval()
all_labels = []
all_preds = []
with torch.no_grad():
    for batch in val_loader:
        inputs = {k: v.to(device) for k, v in batch.items() if k != 'labels'}
        labels = batch['labels'].to(device)
        outputs = model(**inputs)
        preds = torch.argmax(outputs.logits, dim=1)
        all_labels.extend(labels.cpu().numpy())
        all_preds.extend(preds.cpu().numpy())

# Compute metrics
```

```
accuracy = accuracy_score(all_labels, all_preds)
precision, recall, f1, _ = precision_recall_fscore_support(all_labels,␣
 ↪all_preds, average='binary')

print(f'Accuracy: {accuracy*100:.2f}%')
print(f'Precision: {precision*100:.2f}%')
print(f'Recall: {recall*100:.2f}%')
print(f'F1 Score: {f1*100:.2f}%')
```

```
Accuracy: 75.00%
Precision: 76.92%
Recall: 83.33%
F1 Score: 80.00%
```

[80]:
```
# Predict sentiments for a set of sample movie reviews
sample_reviews = [
    "I really enjoyed the movie. The acting was fantastic!",
    "The movie was a waste of time. I wouldn't recommend it to anyone.",
    "The film was creative and surprising.",
    "Absolutely fantastic!",
    "I didn't like the film. It was a mediocre experience.",
]
for review in sample_reviews:
    inputs = tokenizer(review, return_tensors='pt', truncation=True,␣
 ↪padding=True)
    inputs = {k: v.to(device) for k, v in inputs.items()}
    with torch.no_grad():
        outputs = model(**inputs)
    predicted_label = torch.argmax(outputs.logits, dim=1).item()
    sentiment = "Positive" if predicted_label == 1 else "Negative"
    print(f'Text: {review}\nPredicted Sentiment: {sentiment}\n')
```

```
Text: I really enjoyed the movie. The acting was fantastic!
Predicted Sentiment: Positive

Text: The movie was a waste of time. I wouldn't recommend it to anyone.
Predicted Sentiment: Negative

Text: The film was creative and surprising.
Predicted Sentiment: Positive

Text: Absolutely fantastic!
Predicted Sentiment: Positive

Text: I didn't like the film. It was a mediocre experience.
Predicted Sentiment: Negative
```