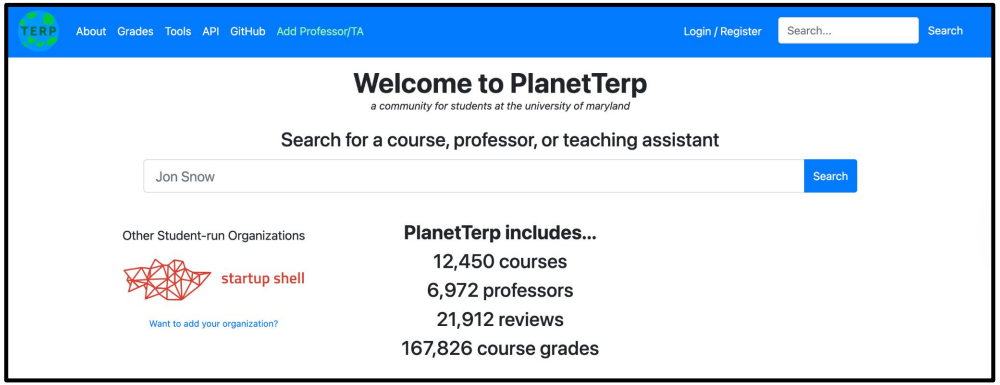
Predicting Professor Review Ratings Using Transformer

CMSC320 - Max Park

Problem Overview

Goal: Given the reviews of 5 professors at UMD, we needed to build a tool that would intake a review and predict the amount of stars the reviewer rated the professor.

Approach: We would use the PlanetTerp API to ingest the reviews of the professors & fine tune a pre-trained model



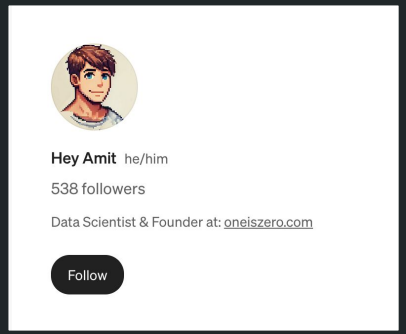
The First Attempt

Going into the Assignment, my approach was:

- Choosing 5 professors with over 100 reviews and not limited to only 1 or 5 stars
- Fine-tuning the Hugging Face DistilBert model to predict the ratings based on user reviews

I borrowed and adapted much of my code from two blogs by 'Hey Amit'.

- Fine-Tuning BERT for Classification: A Practical Guide
- Fine-Tuning DistilBERT: A Step-by-Step Practical Guide



How the Model Was Trained

After retrieving the reviews from Planet Terp API, the data was put into a dataset of:

- Review
- Rating

The ratings were adjusted to ensure that it matched the classification made my DistilBert model.

The dataset was split into training, validation, and testing (for predictions).

Finally, the prompt was **(1)** tokenized, and the **(2)** model, **(3)** training arguments, and **(4)** trainer were copied in with minor edits.

See the corresponding numbers on following slide...

The Architecture

(1)

```
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Preprocessing Function
def preprocess_function(examples):
  '''
  Basially, tokenizes and truncates the prompt and decreases the label by 1 so that it matches up with num_label
  '''
  return tokenizer(examples["prompt"], padding='max_length', truncation=True, max_length=128)

# Apply preprocessing
encoded_dataset = raw_dataset.map(preprocess_function, batched=True)
```

(2)

```
from transformers import AutoModelForSequenceClassification



# Load DistilBERT model for classification
final_model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=5)
print(final_model.config)
```

(3)

```
training_args = TrainingArguments(
    output_dir="./results",           # Directory for saving model checkpoints
    eval_strategy="epoch",     # Evaluate at the end of each epoch
    save_strategy="epoch",
    learning_rate=5e-05,              # Start with a small learning rate
    per_device_train_batch_size=16,  # Batch size per GPU
    num_train_epochs=3,              # Number of epochs
    weight_decay=0.01,               # Regularization
    save_total_limit=2,              # Limit checkpoints to save space
    load_best_model_at_end=True,     # Automatically load the best checkpoint
    logging_dir="./logs",            # Directory for logs
    logging_steps=10,               # Log every 100 steps
    fp16=True                        # Enable mixed precision for faster training
)
```

(4)

```
trainer = Trainer(
    model=final_model,                         # The DistilBERT model
    args=training_args,                        # Training arguments
    train_dataset=encoded_dataset['train'],  # Training data
    eval_dataset=encoded_dataset['validation'],  # Validation data
    tokenizer=tokenizer,
    data_collator=data_collator
)
```

Attempt 1: Trials & Tribulations

```
label
5    586
4    160
3     93
1     93
2     78
Name: count, dtype: int64
professor
Justin Wyss-Gallifent    322
Nelson Padua-Perez       241
Jonathan Fernandes       179
Mestiyage Gunatilleka    168
Clyde Kruskal            100
```

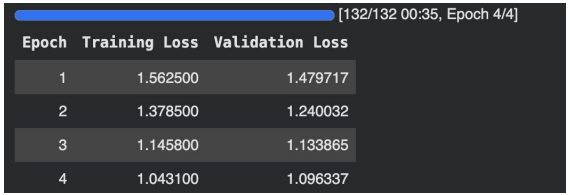
Epoch	Training Loss	Validation Loss
1	1.252700	1.092497
2	0.753900	0.853233
3	0.803200	0.797348

The Results

- Training loss decreased significantly before increasing on the third epoch
- Validation loss during training was low
- Despite the low losses, there was a concern that the high number of 5 star reviews had resulted in a class imbalance in the dataset

Attempt 2: Changing the Dataset to Be More Balanced

```
label
5    230
1    152
4    147
2    108
3    104
Name: count, dtype: int64
professor
Jonathan Fernandes       179
Mestiyage Gunatilleka    168
Timothy Pilachowski      159
Kendall Williams         135
Clyde Kruskal            100
Name: count, dtype: int64
label                    1   2   3   4   5
professor
Clyde Kruskal           28  16  24  17  15
Jonathan Fernandes      26  22  23  41  67
Kendall Williams        22  24  17  29  43
Mestiyage Gunatilleka   11  13  18  36  90
Timothy Pilachowski     65  33  22  24  15
Dataset({
    features: ['professor', 'prompt', 'label'],
    num_rows: 741
})
```



The Approach

- Justin and Nelson who had the highest number of reviews were replaced with Jonathan Fernandes and Mestiyage Gunatilleka who had a lower amount of reviews

The Result

- Higher validation loss and an F1 score of 54%

	precision	recall	f1-score	support
0	0.56	0.87	0.68	23
1	0.29	0.12	0.17	17
2	0.00	0.00	0.00	15
3	0.37	0.45	0.41	22
4	0.67	0.80	0.73	35
accuracy			0.54	112
macro avg	0.38	0.45	0.40	112
weighted avg	0.44	0.54	0.47	112

Attempt 3: Decreased learning rate

```
learning_rate=1e-5,
```

…	precision	recall	f1-score	support
0	0.56	0.78	0.65	23
1	0.27	0.24	0.25	17
2	0.33	0.07	0.11	15
3	0.33	0.41	0.37	22
4	0.69	0.69	0.69	35
accuracy			0.50	112
macro avg	0.44	0.44	0.41	112
weighted avg	0.48	0.50	0.47	112

Approach

- The learning rate was decreased from 5e-5 to 1e-5

Result

- Slightly lower validation loss
- Lower F1 Score for accuracy

[132/132 00:35, Epoch 4/4]

Epoch	Training Loss	Validation Loss
1	0.994900	1.025877
2	0.916600	0.995088
3	0.780100	0.974263
4	0.744100	0.972555

The Final Approach

Finally, the professors used in dataset were adjusted to have a more balanced dataset.

```
label
0    231
4    227
3    167
2    132
1    128
Name: count, dtype: int64
Percentages: label
0    26.10
4    25.65
3    18.87
2    14.92
1    14.46
Name: proportion, dtype: float64
```

```
professor
Jonathan Fernandes    178
Kendall Williams      134
Pedram Sadeghian      134
Christiana Guest      129
James Rainbolt        106
Monique Koppel        104
Clyde Kruskal         100
Name: count, dtype: int64
```

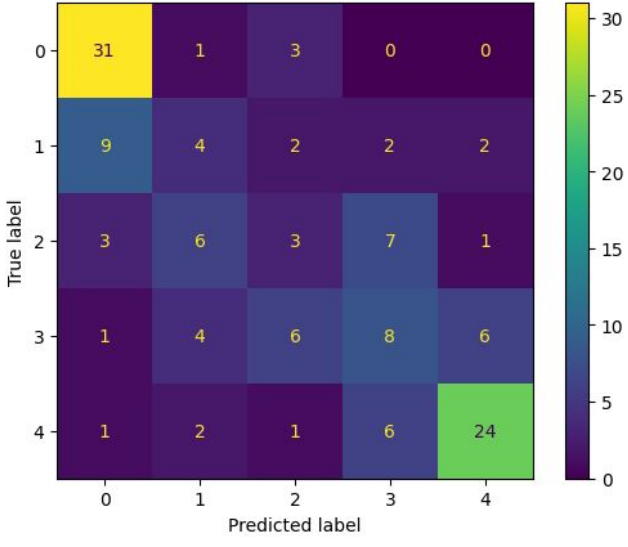
```
Name: count, dtype: int64
label                  0   1   2   3   4
professor
Christiana Guest      62  25  15  11  16
Clyde Kruskal         28  16  24  17  15
James Rainbolt        48  11  12   9  26
Jonathan Fernandes    26  22  23  41  66
Kendall Williams      22  24  17  29  42
Monique Koppel        19  15  25  20  25
Pedram Sadeghian      26  15  16  40  37
```

The Final Results

The Confusion matrix showed the model was good at predicting 0s and 4s (The adjusted 1 and 5), but was average at predicting any other scores.

```
learning_rate=5e-05,
```

Epoch	Training Loss	Validation Loss
1	1.542600	1.344359
2	1.194600	1.177294
3	0.963900	1.110893



The Misclassifications

The Following are examples that the model misclassified

Missed by 1

This woman is literally a mystery. This class was a ridiculous amount of work for a one credit lab, but I gave her an additional star for her ginormous curve at the end of the semester

Predicted: 1. True: 2

Dr. Williams is a solid lecturer. His tests and assignments were fair and not too hard, but could be challenging at times. There was almost no graded homework, but you have to do the suggested problems to do well. His style of teaching is mostly writing down definitions and doing examples, but when he does explain things more, he does well.

Predicted: 5. True: 4

Completely off

OK these reviews are way too dramatic lmao. These kids get to college and learn the hard way you can't BS your way through like high school, then they blame the professor when they fail. Rainbolt is a really underrated professor. I had him for Orgo I. ... His lectures were great... My only complaint about this guy is he definitely is a bit full of himself and gets irritated when students ask questions/come to office hours. He's very impatient when it comes to that kind of stuff and I didn't like that. Other then that great professor.

Predicted: 2. True: 5

Conclusion

The model struggled at correctly predicting ratings between 2-4.
It also struggled with longer reviews with more mixed messaging.
Otherwise, the model seemed to be successful in matching reviews to ratings.

THANK YOU