

# Bonus 2 - Extra Data

In addressing bonus question 2, the goal was to improve the performance of our models by incorporating additional datasets and leveraging existing pre-trained models. Our methodology involved several key steps, each aimed at understanding and optimizing the performance of different models on personality trait prediction from text. For this bonus question, the transformer model is used as the model to evaluate the results on.

## **Dataset Integration and Preprocessing**

We began by processing the original dataset with the additional dataset from Pennebaker and King (1999) which is concerning personality traits. Preprocessing steps included checking the tokenized length of the text to see if the adjusted chunking method was necessary. Moreover, the predictor values were changed to a binary format. This ensured that the merged dataset was coherent and suitable for training our models.

## **BERT Model Enhancement**

We used a BERT model, initially trained on the extended dataset for 5 epochs. Here we observed a promising improvement in performance. As the validation loss was decreasing rapidly, we increased the training epochs to 8 and then to 15. This resulted in a notable enhancement in the model's predictive accuracy, resulting in a micro-averaged F1 score of 0.73 and an ROC AUC of 0.72. These results were very good, as they represented the highest scores achieved in all of our experiments.

## **Use of a Pre-trained personality Huggingface Model**

Next, we utilized a Huggingface model pre-trained specifically for personality trait prediction. By using this we wanted to test if a pre-trained model specifically for this task could improve the performance even more. Initially, this model demonstrated a higher F1 score during a 5-epoch training run. However, extending the training to 8 epochs, even with an adjusted learning rate for finer improvements (we lowered it to  $5e-6$  instead of  $1e-5$ ), did not lead to performance enhancements surpassing our BERT model.

## **Reflections on Model Performance**

The underperformance of the specialized Huggingface model, despite its targeted training, was unexpected. This could indicate a lack of generalizability to new, unseen data, suggesting a potential overfitting issue to its training dataset. Unfortunately, there are no details available of the performance of this model, and therefore cannot fully understand its performance characteristics. What can be noted is that this model has been fine-tuned on BERT base uncased, and our implementation always using the cased version. Although the performance difference cannot be stated from this change alone, it does highlight that potentially the nuances when using capitalized letters makes a difference in this personality trait prediction task.

## **Conclusions**

This approach led to an interesting insight: the base model of BERT, when trained on the increased dataset, outperforms a more specialized model in the same task. This underscores the importance of a model's ability to generalize when dealing with complex tasks such as personality prediction from text.

Through this approach, we gained valuable perspectives on task-specific pre-trained models, the crucial role of having more data available, and model generalizability. .

## References

Pennebaker, J. W., & King, L. A. (1999). Linguistic styles: language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6), 1296–1312.  
<https://doi.org/10.1037/0022-3514.77.6.1296>

# Results

BERT two datasets

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.680700	0.678448	0.587056	0.573815	0.051185
2	0.642800	0.657924	0.629830	0.603865	0.072986
4	0.597900	0.647590	0.625550	0.620769	0.097630

```
1 training_args = TrainingArguments(
2     output_dir='./',
3     logging_dir='./',
4     per_device_train_batch_size=16,
5     per_device_eval_batch_size=16,
6     num_train_epochs=4,
7     weight_decay=0.01,
8     validation_strategy='epoch',
9     logging_dir='./',
10    logging_steps=10,
11    warmup_steps=5,
12    evaluation_strategy='epoch',
13    save_strategy='epoch',
14    disable_tqdm=False,
15    load_best_model_at_end=True,
16    push_to_hub=False,
17    gradient_accumulation_steps=4,
18    fp16_backend='auto',
19    torch_compile_backend='inductor',
20    torch_compile_mode='reduce-overhead',
21)
22
23 trainer = GradientDescentTrainer(
24     model=model,
25     args=training_args,
26     data_loader=train_data_loader,
27     eval_loader=eval_data_loader,
28     data_collator=data_collator,
29     compute_metrics=compute_metrics,
30)
31
32 trainer.train()
```

Bert 8 epochs

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.583700	0.642112	0.630580	0.639135	0.107583
2	0.491100	0.618321	0.695629	0.674136	0.157820
4	0.439300	0.601418	0.701469	0.697573	0.191469
6	0.396700	0.590451	0.720304	0.713467	0.229858
7	0.379500	0.587777	0.724084	0.714801	0.231280

```
1 training_args = TrainingArguments(
2     output_dir='./',
3     logging_dir='./',
4     per_device_train_batch_size=16,
5     per_device_eval_batch_size=16,
6     num_train_epochs=8,
7     weight_decay=0.01,
8     validation_strategy='epoch',
9     logging_dir='./',
10    logging_steps=10,
11    warmup_steps=5,
12    evaluation_strategy='epoch',
13    save_strategy='epoch',
14    disable_tqdm=False,
15    load_best_model_at_end=True,
16    push_to_hub=False,
17    gradient_accumulation_steps=4,
18    fp16_backend='auto',
19    torch_compile_backend='inductor',
20    torch_compile_mode='reduce-overhead',
21)
22
23 trainer = GradientDescentTrainer(
24     model=model,
25     args=training_args,
26     data_loader=train_data_loader,
27     eval_loader=eval_data_loader,
28     data_collator=data_collator,
29     compute_metrics=compute_metrics,
30)
31
32 trainer.train()
```

## Bert 15 epochs

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.680700	0.679851	0.583576	0.552164	0.036493
2	0.629100	0.656964	0.637448	0.600673	0.078199
4	0.548700	0.631546	0.654656	0.643514	0.114692
6	0.480100	0.610031	0.674069	0.668878	0.156872
8	0.448500	0.597665	0.714930	0.693082	0.197156
10	0.394300	0.585504	0.721156	0.710439	0.223697
12	0.365200	0.583853	0.731721	0.723183	0.248815
14	0.359500	0.583148	0.733248	0.723364	0.249289

```

27     predictions=preds,
28     labels=p.label_ids)
29     return result

1 training_args = TrainingArguments(
2     output_dir=output_dir,
3     learning_rate=5e-5,
4     per_device_train_batch_size=8,
5     per_device_eval_batch_size=16,
6     num_train_epochs=15,
7     weight_decay=0.01,
8     warmup_steps = 20,
9     logging_steps = 8,
10    evaluation_strategy="epoch",
11    save_strategy="epoch",
12    disable_tqdm = False,
13    load_best_model_at_end=True,
14    push_to_hub=False,
15    gradient_accumulation_steps=8,
16    gradient_checkpointing=True,
17    fp16=True,
18 )
19 trainer = Trainer(
20     model=model,
21     args=training_args,
22     train_dataset=train_dataset,
23     eval_dataset=test_dataset,
24     tokenizer=tokenizer,
25     data_collator=data_collator,
26     compute_metrics=compute_metrics,
27 )
28
29 trainer.train()

```

/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:4  
 warnings.warn( [21152115.4336, Epoch 14/15]

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.680700	0.679851	0.583576	0.552164	0.036493
2	0.629100	0.656964	0.637448	0.600673	0.078199
4	0.548700	0.631546	0.654656	0.643514	0.114692
6	0.480100	0.610031	0.674069	0.668878	0.156872
8	0.448500	0.597665	0.714930	0.693082	0.197156
10	0.394300	0.585504	0.721156	0.710439	0.223697
12	0.365200	0.583853	0.731721	0.723183	0.248815
14	0.359500	0.583148	0.733248	0.723364	0.249289

## Minej base

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.676600	0.677387	0.637700	0.561451	0.055342

1	0.661600	0.665066	0.631056	0.590349	0.078922
2	0.632800	0.657511	0.632188	0.607283	0.087103
3	0.612900	0.653189	0.647121	0.609916	0.098171
4	0.604400	0.651181	0.647033	0.615368	0.100577

```

1 training_args = TrainingArguments(
2     output_dir=output_dir,
3     learning_rate=1e-5,
4     per_device_train_batch_size=8,
5     per_device_eval_batch_size=16,
6     num_train_epochs=5,
7     weight_decay=0.01,
8     warmup_steps = 20,
9     logging_steps = 8,
10    evaluation_strategy="epoch",
11    save_strategy="epoch",
12    disable_tqdm = False,
13    load_best_model_at_end=True,
14    push_to_hub=False,
15    gradient_accumulation_steps=8,
16    gradient_checkpointing=True,
17    fp16=True,
18 )
19 trainer = Trainer(
20     model=model,
21     args=training_args,
22     train_dataset=train_dataset,
23     eval_dataset=test_dataset,
24     tokenizer=tokenizer,
25     data_collator=data_collator,
26     compute_metrics=compute_metrics,
27 )
28
29 trainer.train()

```

/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py: warnings.warn(

[695/695 14:36, Epoch 4/5]

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.676800	0.677387	0.637700	0.561451	0.055342
1	0.661600	0.665066	0.631056	0.590349	0.078922
2	0.632800	0.657511	0.632188	0.607283	0.087103
3	0.612900	0.653189	0.647121	0.609916	0.098171
4	0.604400	0.651181	0.647033	0.615368	0.100577

Minej 8 epochs

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.603200	0.650346	0.630122	0.610806	0.100577
1	0.544500	0.632890	0.665636	0.645080	0.129451
2	0.519100	0.624866	0.684726	0.657953	0.152551
3	0.486200	0.628331	0.690597	0.665036	0.155919
4	0.487400	0.620186	0.690289	0.668171	0.158325
5	0.464100	0.625740	0.702137	0.666333	0.158807
6	0.440000	0.612493	0.707481	0.679072	0.179018
7	0.445500	0.614774	0.709467	0.680666	0.186718

```
1 training_args = TrainingArguments(  
2     output_dir=output_dir,  
3     learning_rate=1e-5,  
4     per_device_train_batch_size=8,  
5     per_device_eval_batch_size=16,  
6     num_train_epochs=8,  
7     weight_decay=0.01,  
8     warmup_steps = 20,  
9     logging_steps = 8,  
10    evaluation_strategy="epoch",  
11    save_strategy="epoch",  
12    disable_tqdm = False,  
13    load_best_model_at_end=True,  
14    push_to_hub=False,  
15    gradient_accumulation_steps=8,  
16    gradient_checkpointing=True,  
17    fp16=True,  
18 )  
19 trainer = Trainer(  
20     model=model,  
21     args=training_args,  
22     train_dataset=train_dataset,  
23     eval_dataset=test_dataset,  
24     tokenizer=tokenizer,  
25     data_collator=data_collator,  
26     compute_metrics=compute_metrics,  
27 )  
28  
29 trainer.train()
```

/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:42  
warnings.warn(  
[1112/1112 23:03, Epoch 7/8]

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.603200	0.650346	0.630122	0.610806	0.100577
1	0.544500	0.632890	0.665636	0.645080	0.129451
2	0.519100	0.624866	0.684726	0.657953	0.152551
3	0.486200	0.628331	0.690597	0.665036	0.155919
4	0.487400	0.620186	0.690289	0.668171	0.158325
5	0.464100	0.625740	0.702137	0.666333	0.158807
6	0.440000	0.612493	0.707481	0.679072	0.179018
7	0.445500	0.614774	0.709467	0.680666	0.186718

Minaj 8 epochs but adjusted LR to 5e-5

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.682900	0.680691	0.636227	0.552853	0.053468
1	0.669900	0.673964	0.630855	0.567297	0.064066
2	0.667700	0.667446	0.634642	0.584609	0.072254

3	0.654300	0.664436	0.633788	0.589001	0.069846
4	0.639200	0.662224	0.637995	0.591047	0.073699
5	0.639600	0.661599	0.640868	0.592162	0.077071
6	0.620400	0.658707	0.636593	0.595689	0.076108
7	0.620700	0.659982	0.640544	0.595281	0.077553

```

1 training_args = TrainingArguments(
2     output_dir=output_dir,
3     learning_rate=5e-6,
4     per_device_train_batch_size=8,
5     per_device_eval_batch_size=16,
6     num_train_epochs=8,
7     weight_decay=0.01,
8     warmup_steps = 20,
9     logging_steps = 8,
10    evaluation_strategy="epoch",
11    save_strategy="epoch",
12    disable_tqdm = False,
13    load_best_model_at_end=True,
14    push_to_hub=False,
15    gradient_accumulation_steps=8,
16    gradient_checkpointing=True,
17    fp16=True,
18 )
19 trainer = Trainer(
20     model=model,
21     args=training_args,
22     train_dataset=train_dataset,
23     eval_dataset=test_dataset,
24     tokenizer=tokenizer,
25     data_collator=data_collator,
26     compute_metrics=compute_metrics,
27 )
28
29 trainer.train()

```

/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py: warnings.warn( [1112/1112 22:44, Epoch 7/8]

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.682900	0.680691	0.636227	0.552853	0.053468
1	0.669900	0.673964	0.630855	0.567297	0.064066
2	0.667700	0.667446	0.634642	0.584609	0.072254
3	0.654300	0.664436	0.633788	0.589001	0.069846
4	0.639200	0.662224	0.637995	0.591047	0.073699
5	0.639600	0.661599	0.640868	0.592162	0.077071
6	0.620400	0.658707	0.636593	0.595689	0.076108
7	0.620700	0.659982	0.640544	0.595281	0.077553