UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP - HW3

Student name: *Dimitrios Chrysos*
sdi: 2100275

Course: *Artificial Intelligence II (ΥΣ19)*
Semester: *Spring Semester 2024-2025*

Contents

1. Abstract

1.1. Task

The assignment is to build a sentiment classifier using two pretrained transformer models (BERT and DistilBERT) on a Twitter sentiment dataset. The models must be fine-tuned using PyTorch, and the performance must be evaluated via different experiments and metrics. The project is written in Python for a given English language Twitter dataset. Three datasets will be used: `train_dataset`,`val_dataset`, and `test_dataset`, which are used for training, validation, and testing, respectively.

1.2. Notebooks

The following notebooks were developed as part of Assignment 3

1. BERT - Kaggle Notebook
2. DistilBERT - Kaggle Notebook

2. Data processing and analysis

2.1. Pre-processing

- For BERT, two different preprocessing techniques were used across the experiments. Initially (up to Experiment 5), the same preprocessing steps from the previous assignments were applied. However, starting with Experiment 6, a revised and more targeted preprocessing approach was introduced to improve model performance while avoiding over-normalization and preserving sentiment-rich slang and abbreviations.
- DistilBERT uses only the updated preprocessing function.
- This updated preprocessing now performs the following operations:
  1. Lowercases the text
  2. Removes URLs
  3. Removes mentions
  4. Keeps hashtags (removes only the # symbol)
  5. Removes non-ASCII characters
  6. Removes excess spaces
  7. Strips leading and trailing whitespace

2.2. Data partitioning for train, test and validation

- The data was already portioned.

2.3. Vectorization

- For both the BERT and DistilBERT models, vectorization is done through pre-trained tokenizers provided by HuggingFace: BertTokenizer and DistilBertTokenizer, respectively.
- These tokenizers transform raw text into a format suitable for transformer models by:
  - Converting words and subwords into token IDs based on each model's vocabulary.
  - Creating attention masks that help the model distinguish between actual tokens and padding.
  - Performing subword tokenization, which handles unknown words by breaking them into smaller known units.
  - Automatically handling special tokens (like [CLS], [SEP]).
- The maximum sequence length is discovered by tokenizing all sentences in the training, validation, and test sets with special tokens ([CLS] and [SEP]) and identifying the longest one. The value is then used during tokenization to ensure all sequences are uniformly padded to the same length.
- This process ensures that both models receive input in the correct numerical format and structure required for fine-tuning.

3. Algorithms and Experiments

3.1. Experiments - BERT

1. First Experiment - Instructor's parameters
   - The purpose of the first experiment was to obtain an initial understanding of the evaluation metrics.
   - For the above reason, the parameters used were the ones provided in the instructor's example code. More precisely, the experiment used the following configuration:
     - `batch_size = 32`
     - `epochs = 4`
     - `learning_rate = 2e-5`
     - `eps = 1e-8`
     - `num_warmup_steps = 0`

- Results:

```
Validation Accuracy: 0.8362

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.84      0.84     21197
           1       0.84      0.83      0.84     21199

    accuracy                           0.84     42396
   macro avg       0.84      0.84      0.84     42396
weighted avg       0.84      0.84      0.84     42396
```

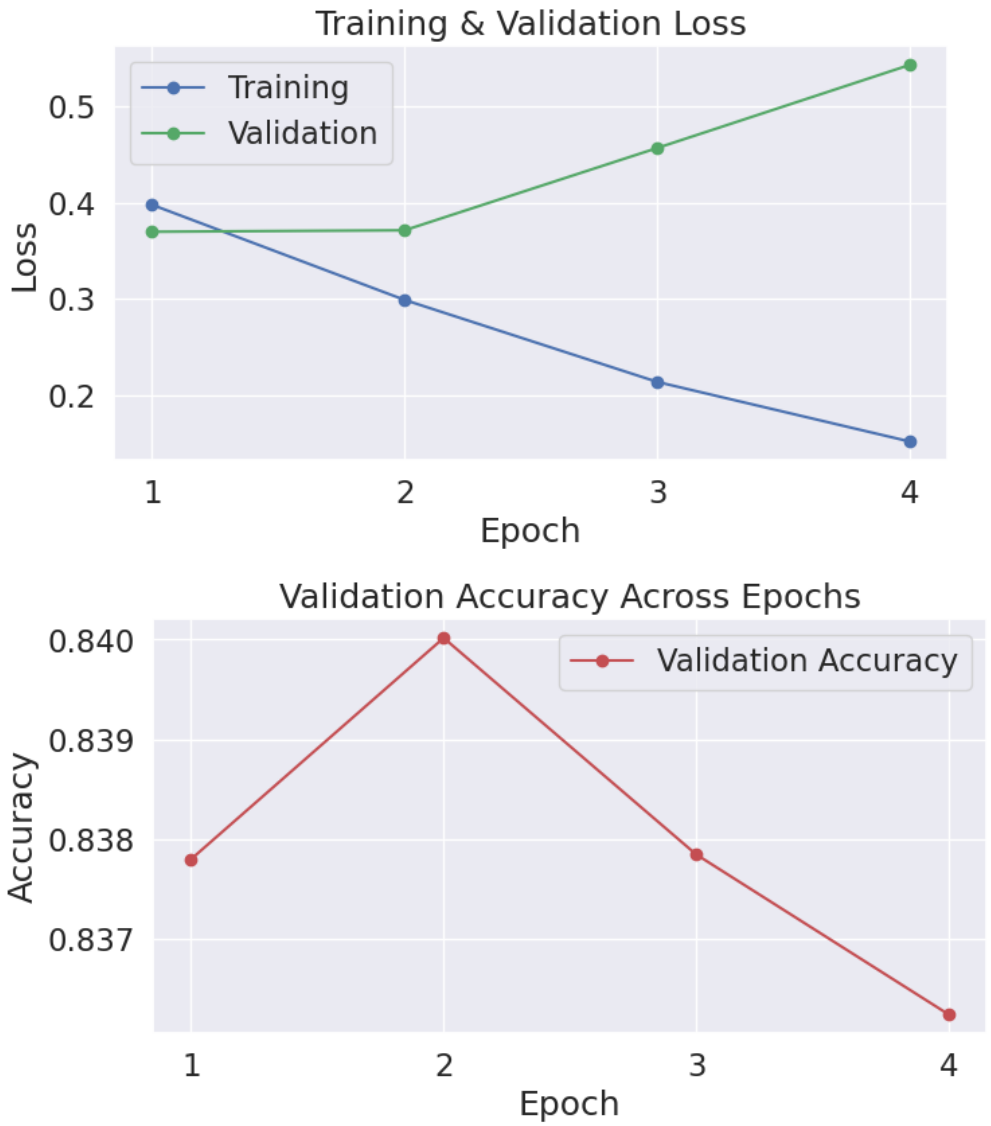


Figure 1: Experiment 1

- Time needed: `0:56:46 (h:mm:ss)`
- Some initial observations indicate that the **model peaks** at the **second epoch**.
- In both the **Training & Validation Loss** and the **Validation Accuracy Across Epochs** plots, the validation lines worsen after epoch two. Especially in the

first plot, the overfitting is obvious since the loss for validation increases while the loss for training decreases.

2. Second Experiment - Optuna

   - The authors of BERT paper (Appendix A.3), recommend choosing the training hyperparameters from the following values:
     - `Batch size: 16, 32`
     - `Learning rate (Adam): 5e-5, 3e-5, 2e-5`
     - `Number of epochs: 2, 3, 4`
   - For the above reason, the Optuna suggestions are exactly the ones described above.
   - This experiment runs a study for `batch_size = 32` and the values listed above for `learning_rate` and `epochs`.
   - Epsilon parameter `eps = 1e-8` is "a very small number to prevent any division by zero in the implementation" (from here).
   - Due to lengthy training times, a 20% sample of the original training data was used for this experiment.
   - GridSampler exhaustively explores the search space by systematically generating every possible combination of specified hyperparameter values, ensuring that each combination is tried exactly once with no repetition or randomness involved.
   - Results:

	Trial	Validation Loss	params_epoch	params_learning_rate
0	0	0.78	4	3.00e-05
1	1	0.41	2	2.00e-05
2	2	0.42	2	3.00e-05
3	3	0.54	3	3.00e-05
4	4	0.61	4	2.00e-05
5	5	0.90	4	5.00e-05
6	6	0.62	3	5.00e-05
7	7	0.43	2	5.00e-05
8	8	0.47	3	2.00e-05

Figure 2: Experiment 2

   - Time needed: `1:39:07 (h:mm:ss)`
   - Based on the results above, training for 2 epochs consistently yields the best performance, with **Trial 1** achieving the best score (lowest `Validation Loss`).

3. Third Experiment - Try Optuna results

   - This experiment's only purpose is to generate more advanced plots and metrics with the configuration from the results of the previous experiment.

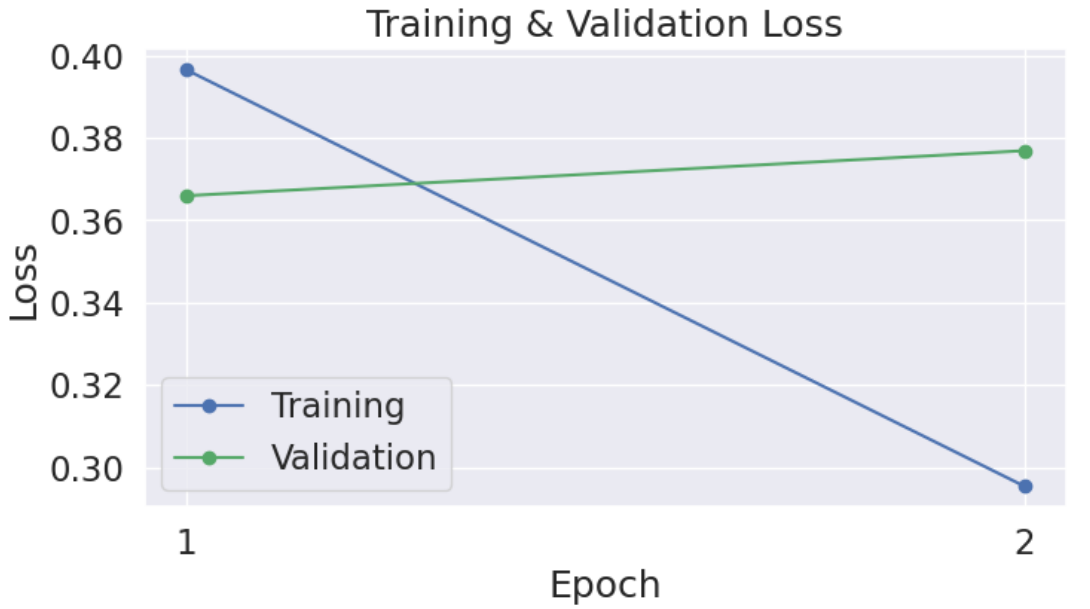
- Configuration:
  - `Batch size: 32`
  - `Number of epochs: 2`
  - `Learning rate: 2e-5`
  - `eps = 1e-8`
  - `num_warmup_steps = 0`
- Results:

```
Validation Accuracy: 0.8432

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.84     21197
           1       0.85      0.83      0.84     21199

    accuracy                           0.84     42396
   macro avg       0.84      0.84      0.84     42396
weighted avg       0.84      0.84      0.84     42396
```



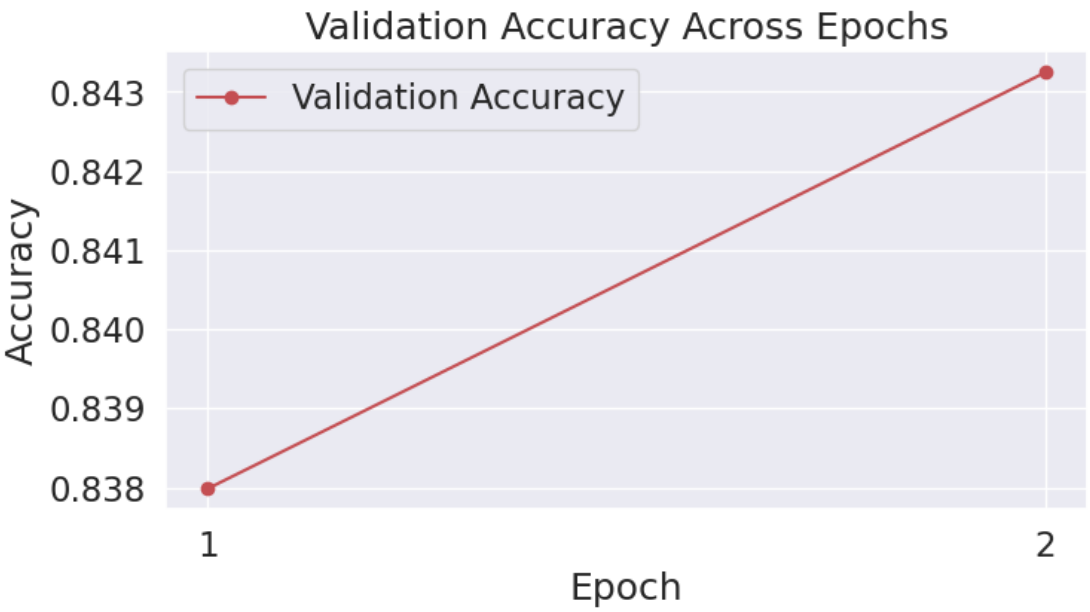


Figure 3: Experiment 3

- Time needed: `0:28:15 (h:mm:ss))`
- This experiment demonstrates no indications of overfitting and achieves an accuracy of 0.8432, beating all metrics of Experiment 1.

- For the above reason, Experiment 3 is considered the best one so far.

4. Fourth Experiment - Experiment 3 with smaller batch size

   - This experiment's purpose is to understand if we can have any improvement using a smaller batch size.
   - It uses `batch_size = 16`, the other batch size value recommended by the authors of BERT paper (Appendix A.3).
   - It also increased **epochs** to three, since a smaller batch size usually reduces overfitting and this can potentially score better results in later epochs.
   - Configuration:
     - `Batch size: 16`
     - `Number of epochs: 3`
     - `Learning rate: 2e-5`
     - `eps = 1e-8`
     - `num_warmup_steps = 0`
   - Results:

```
Validation Accuracy: 0.8378

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.85      0.84     21197
           1       0.85      0.83      0.84     21199

    accuracy                           0.84     42396
   macro avg       0.84      0.84      0.84     42396
weighted avg       0.84      0.84      0.84     42396
```

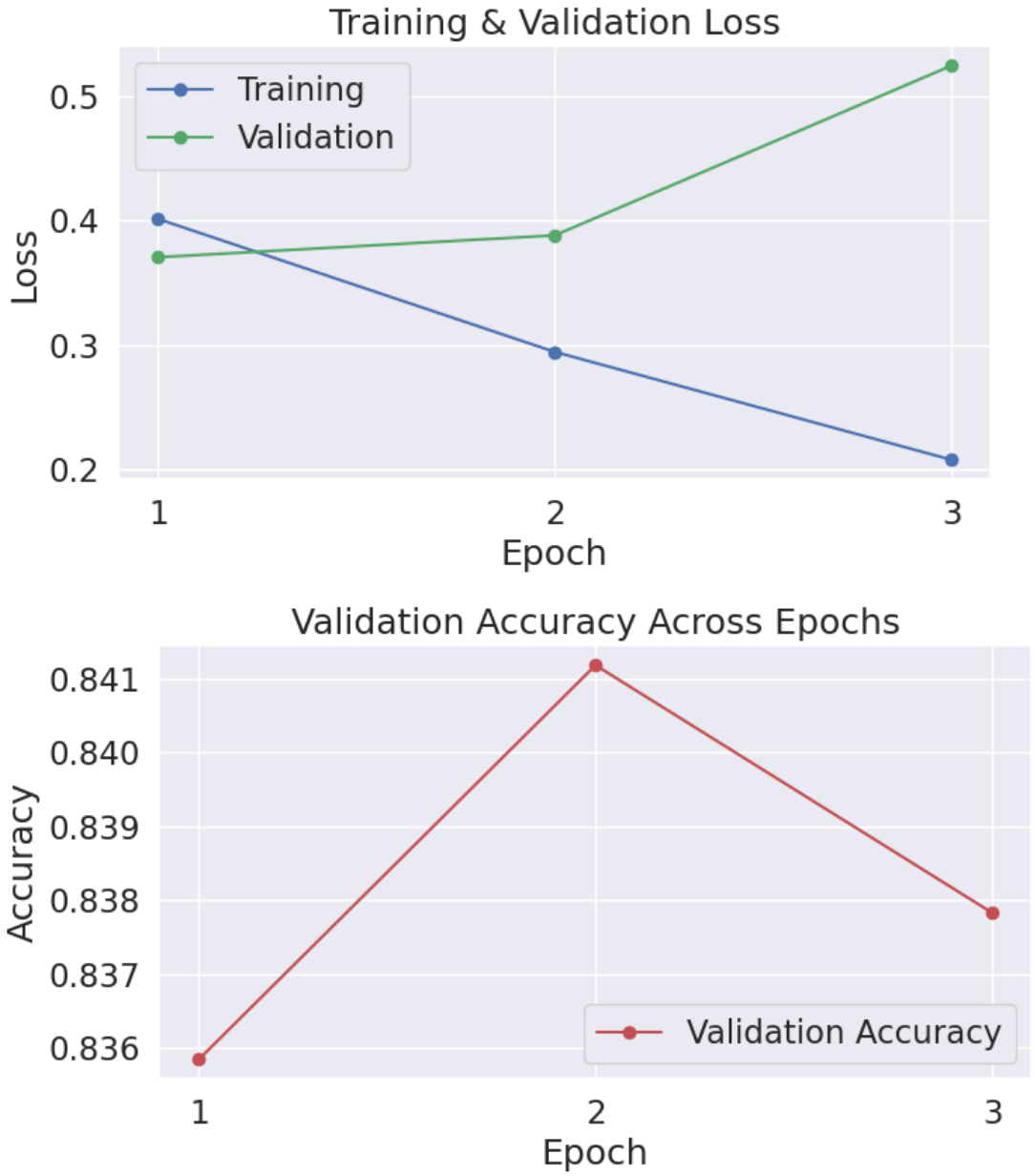


Figure 4: Experiment 4

- Time needed: `0:51:43 (h:mm:ss))`
- The results indicate that there is no real improvement with a smaller batch size.

- The overfitting remains similar to that of Experiment 1.
- The validation accuracy plot reveals a decrease in the accuracy in epoch 3 compared to epoch 2, and epoch 2 in this experiment has a smaller value than that of experiment 2.
- The validation loss plot exhibits a similar pattern, showing no improvement in performance.
- Time needed is also almost double that of the previous experiment.
- For the above reasons **Experiment 3** is considered the best so far.

5. Fifth Experiment - Try warm-up steps

   - This experiment's purpose is to understand if adding some warm-up steps can increase the performance of the model.
   - Bert models are pretrained, so jumping in with full learning rate right away can destabilize fine-tuning. A small warm-up (e.g., 0–10% of total steps) helps make training smoother and more effective. This works by linearly increasing the learning rate from 0 to the target learning rate for the first `num_warmup_steps`.
   - Configuration:
     - `Batch size: 32`
     - `Number of epochs: 2`
     - `Learning rate: 2e-5`
     - `eps = 1e-8`
     - `total_steps = len(train_dataloader) * epochs`
     - `num_warmup_steps = int(0.05 * total_steps)`
   - Results:

```
Validation Accuracy: 0.8431

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.84     21197
           1       0.85      0.84      0.84     21199

    accuracy                           0.84     42396
   macro avg       0.84      0.84      0.84     42396
weighted avg       0.84      0.84      0.84     42396
```

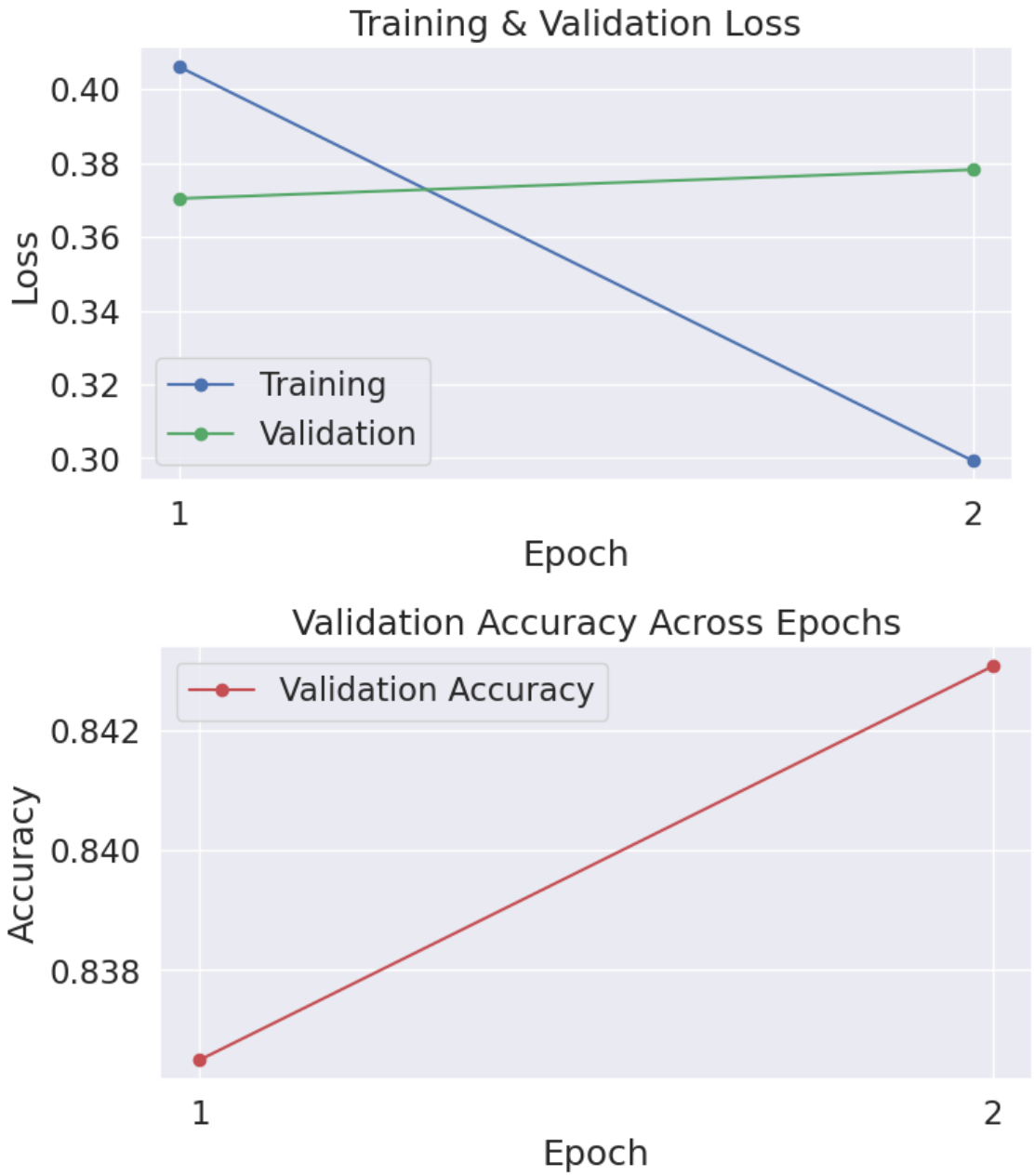


Figure 5: Experiment 5

- Time needed: `0:28:22 (h:mm:ss)`
- The results are almost identical to Experiment 3, for this reason, we will keep

considering **Experiment 3** as the best one so far.

6. Sixth Experiment - New preprocessing function

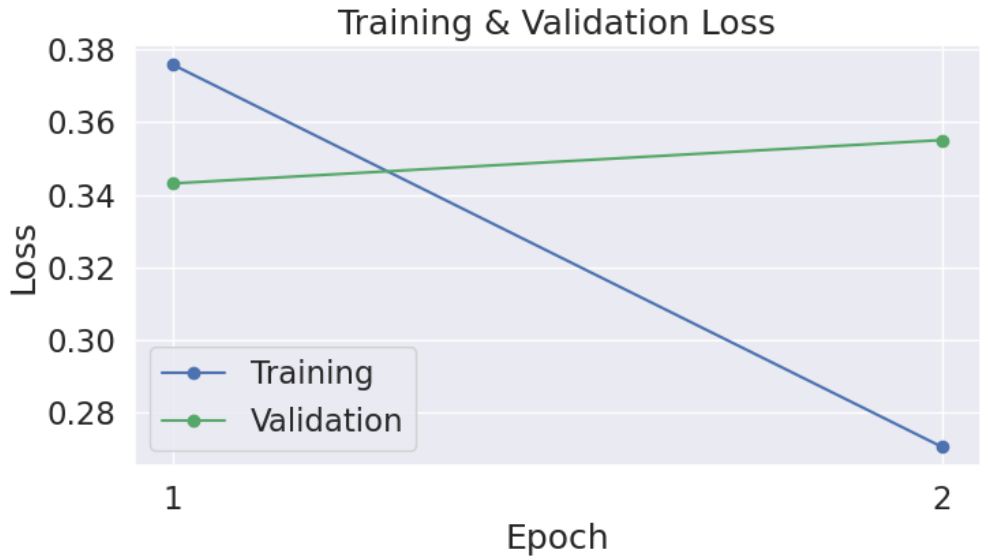
   - The previous experiments produced similar results, prompting a shift in strategy. This experiment introduces a different preprocessing approach to overcome stagnation.
   - Now, preprocessing only does the following procedures:
     - Lowercases the text
     - Removes URLs
     - Removes mentions
     - Keeps hashtags (removes only '#')
     - Removes non-ASCII
     - Removes excess spaces
     - Strips the text
   - The rest of the configuration is the same as Experiment 3.
   - Results:

```
Validation Accuracy: 0.8535

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.86      0.85     21197
           1       0.86      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```



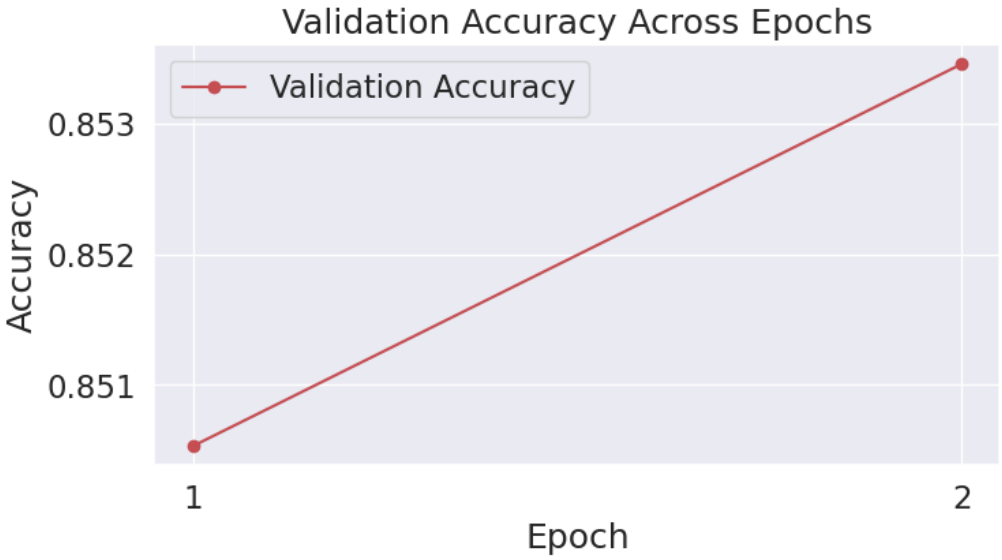


Figure 6: Experiment 6

- Time needed: `1:01:26 (h:mm:ss)`
- The results indicate clear performance improvement in all metrics. Accuracy impressively seems to have increased by 1%, while the validation loss is close to 2% lower.
- For the above reason, this experiment is considered the best so far.

7. Seventh Experiment - Optuna Extra Parameters

   - For this study, the configuration is a bit different than that of Experiment 2.
   - More precisely, the study consists of 20 runs with a configuration for each one created by a selection of the following values per parameter.
     - `learning_rate: 5e-5, 3e-5, 2e-5`
     - `epochs: 2, 3`
     - `hidden_dropout: 0.1, 0.2, 0.3`
     - `attention_dropout: 0.1, 0.2, 0.3`
     - `weight_decay: 0.0, 0.1, 0.2, 0.3`
   - This experiment runs a study for `batch_size = 32`, `eps = 1e-8` and a 20% sample of the original training data.
   - Results:

	Trial	Validation Loss	params_attention_dropout	params_epoch	params_hidden_dropout	params_learning_rate	params_weight_decay
0	0	0.418070	0.2	3	0.3	0.00005	0.0
1	1	0.503414	0.3	3	0.1	0.00005	0.2
2	2	0.393722	0.1	2	0.1	0.00002	0.2
3	3	0.390611	0.3	2	0.1	0.00003	0.3
4	4	0.400425	0.2	2	0.3	0.00005	0.2
5	5	0.447955	0.3	3	0.1	0.00003	0.2
6	6	0.418584	0.2	3	0.2	0.00002	0.3
7	7	0.425632	0.3	3	0.3	0.00005	0.2
8	8	0.430293	0.2	3	0.3	0.00005	0.3
9	9	0.469454	0.1	3	0.1	0.00002	0.0
10	10	0.398464	0.2	2	0.3	0.00002	0.1
11	11	0.391243	0.2	2	0.1	0.00002	0.2
12	12	0.401370	0.1	2	0.2	0.00005	0.1
13	13	0.587362	0.1	3	0.1	0.00005	0.1
14	14	0.401583	0.2	3	0.3	0.00002	0.0
15	15	0.444725	0.3	3	0.1	0.00003	0.2
16	16	0.438241	0.1	3	0.3	0.00005	0.1
17	17	0.481473	0.2	3	0.2	0.00005	0.1
18	18	0.421897	0.3	3	0.2	0.00003	0.3
19	19	0.413096	0.3	3	0.3	0.00002	0.0

Figure 7: Experiment 7

   - Time needed: `7:10:32 (h:mm:ss)`
   - Best trial is trial 3 with a validation loss of 0.390611 and the configuration:
     - `learning_rate: 3e-5`
     - `epochs: 2`
     - `hidden_dropout: 0.1`
     - `attention_dropout: 0.3`

  - `weight_decay: 0.3`

8. Eighth Experiment - Try Optuna results

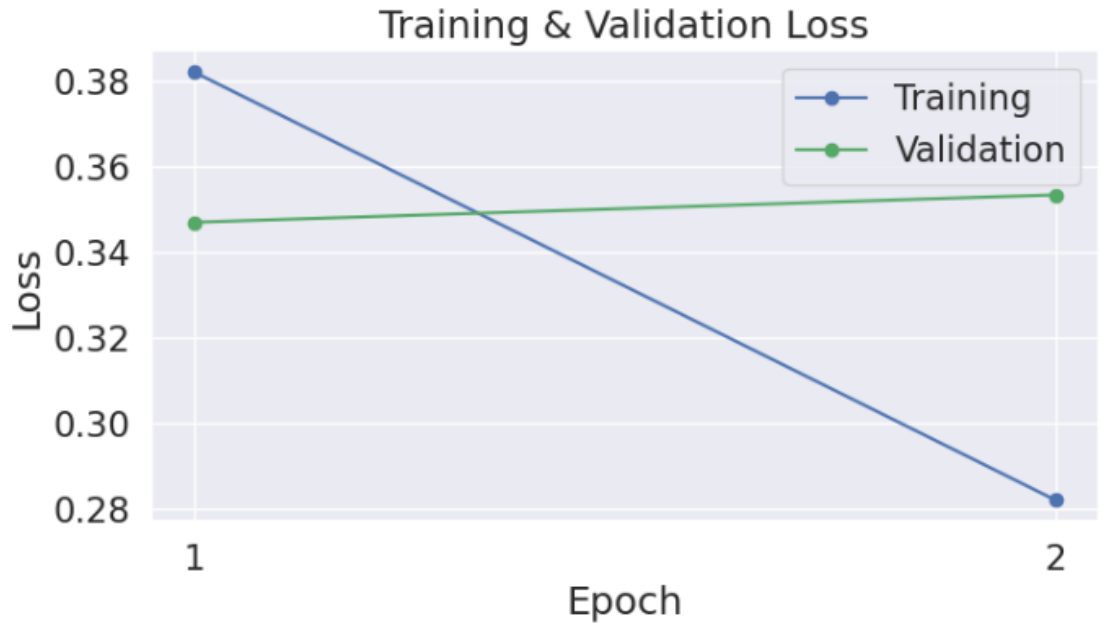
   - This experiment's only purpose is to generate more advanced plots and metrics with the configuration from the results of the previous experiment.
   - Configuration:
     - `Batch size: 32`
     - `Number of epochs: 2`
     - `Learning rate: 3e-5`
     - `eps = 1e-8`
     - `num_warmup_steps = 0`
     - `hidden_dropout: 0.1`
     - `attention_dropout: 0.3`
     - `weight_decay: 0.3`
   - Results:

```
Validation Accuracy: 0.8547

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.86      0.86     21197
           1       0.86      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```



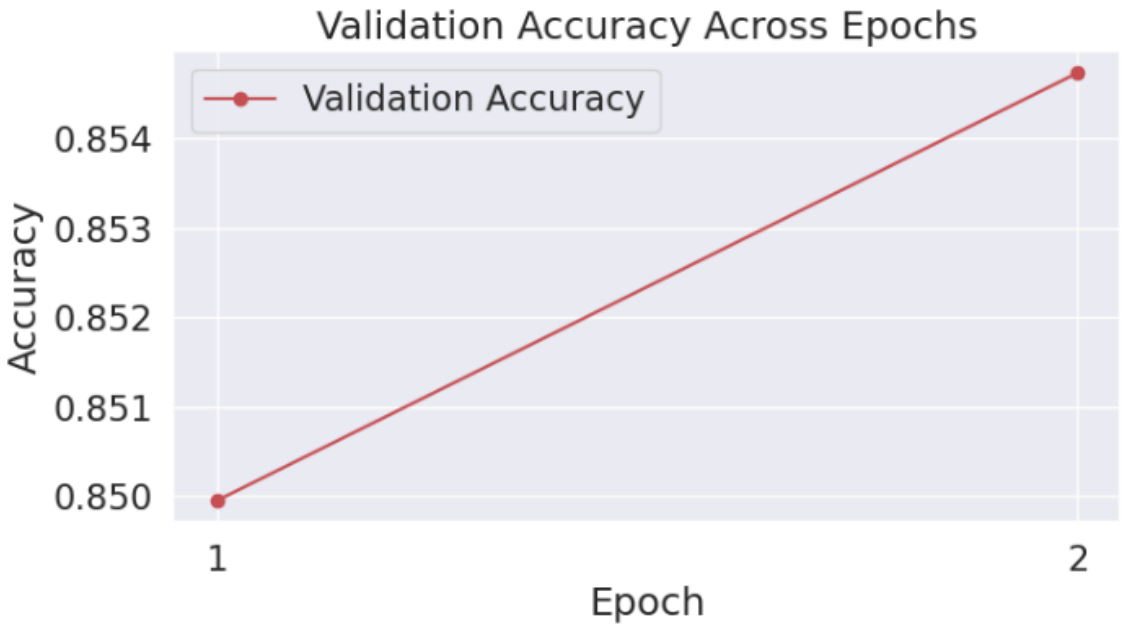


Figure 8: Experiment 8

- Time needed: `1:01:35 (h:mm:ss)`

- The above results are the best so far, more precisely:
  - Validation Accuracy is higher than that of Experiment 6.
  - Classification Report is almost identical, but this Experiment has a bit better f1-score.
  - Training & Validation Loss plot shows smaller overfitting compared to Experiment 6, while preserving the validation loss value of that experiment.
  - Validation Accuracy Across Epochs is also higher.
- For the above reasons **Experiment 8** is considered the best performing.

3.2. Experiments - DistilBERT

Based on the insights gained from the BERT model experiments, the DistilBERT experiments were conducted with a more targeted approach. More specifically, the experimentation for DistilBERT used only the new preprocessing function.

1. First Experiment - Try BERT's fine-tuned parameters
   - The first experiment for DistilBERT model uses the configuration of the best experiment for the BERT model.
   - Configuration:
     - `Batch size: 32`
     - `Number of epochs: 2`
     - `Learning rate: 3e-5`
     - `eps = 1e-8`
     - `num_warmup_steps = 0`
     - `hidden_dropout: 0.1`
     - `attention_dropout: 0.3`
     - `weight_decay: 0.3`
   - Results:

```
Validation Accuracy: 0.8471

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85     21197
           1       0.85      0.84      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

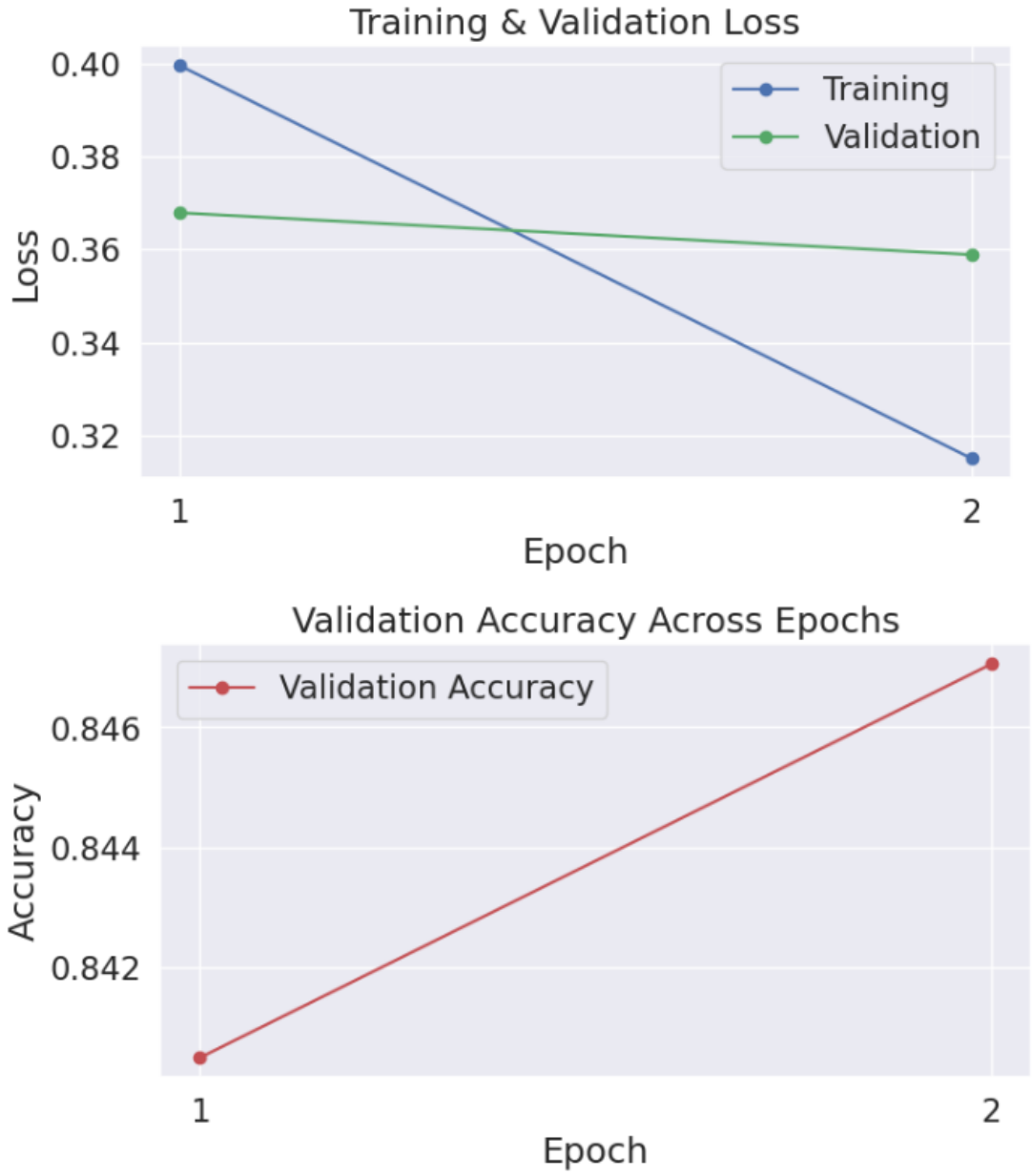


Figure 9: Experiment 1

- Time needed: `0:31:16 (h:mm:ss)`
- Compared to the BERT model, this model performs worse across all evaluation metrics when using the same hyper-parameters, but it achieves approx-

imately half the training time.

2. Second Experiment - Optuna

   - The second experiment is a study with Optuna to find the optimal parameters for this model.
   - More precisely, the study consists of 20 runs with a configuration for each one created by a selection of the following values per parameter.
     - `learning_rate: 5e-5, 3e-5, 2e-5`
     - `epochs: 2, 3`
     - `hidden_dropout: 0.1, 0.2, 0.3`
     - `attention_dropout: 0.1, 0.2, 0.3`
     - `weight_decay: 0.0, 0.1, 0.2, 0.3`
   - This experiment runs a study for `batch_size = 32`, `eps = 1e-8` and a 20% sample of the original training data.
   - Results:

	Trial	Validation Loss	params_attention_dropout	params_epoch	params_hidden_dropout	params_learning_rate	params_weight_decay
0	0	0.433325	0.2	3	0.3	0.00005	0.0
1	1	0.490467	0.3	3	0.1	0.00005	0.2
2	2	0.398665	0.1	2	0.1	0.00002	0.2
3	3	0.402213	0.3	2	0.1	0.00003	0.3
4	4	0.405834	0.2	2	0.3	0.00005	0.2
5	5	0.428273	0.3	3	0.1	0.00003	0.2
6	6	0.414444	0.2	3	0.2	0.00002	0.3
7	7	0.429372	0.3	3	0.3	0.00005	0.2
8	8	0.431017	0.2	3	0.3	0.00005	0.3
9	9	0.438741	0.1	3	0.1	0.00002	0.0
10	10	0.406745	0.2	2	0.3	0.00002	0.1
11	11	0.397118	0.2	2	0.1	0.00002	0.2
12	12	0.415624	0.1	2	0.2	0.00005	0.1
13	13	0.542146	0.1	3	0.1	0.00005	0.1
14	14	0.405825	0.2	3	0.3	0.00002	0.0
15	15	0.427532	0.3	3	0.1	0.00003	0.2
16	16	0.441479	0.1	3	0.3	0.00005	0.1
17	17	0.452253	0.2	3	0.2	0.00005	0.1
18	18	0.416736	0.3	3	0.2	0.00003	0.3
19	19	0.414508	0.3	3	0.3	0.00002	0.0

Figure 10: Experiment 2

   - Time needed: `3:38:13 (h:mm:ss)`
   - Best trial is trial 11 with a validation loss of 0.397118 and the configuration:
     - `learning_rate: 2e-5`
     - `epochs: 2`

  - `hidden_dropout: 0.1`
  - `attention_dropout: 0.2`
  - `weight_decay: 0.2`

3. Third Experiment - Try Optuna Results

   - This experiment's only purpose is to generate more advanced plots and metrics with the configuration from the results of the previous experiment.
   - Configuration:
     - `Batch size: 32`
     - `Number of epochs: 2`
     - `Learning rate: 2e-5`
     - `eps = 1e-8`
     - `num_warmup_steps = 0`
     - `hidden_dropout: 0.1`
     - `attention_dropout: 0.2`
     - `weight_decay: 0.2`
   - Results:

```
Validation Accuracy: 0.8468

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85     21197
           1       0.85      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

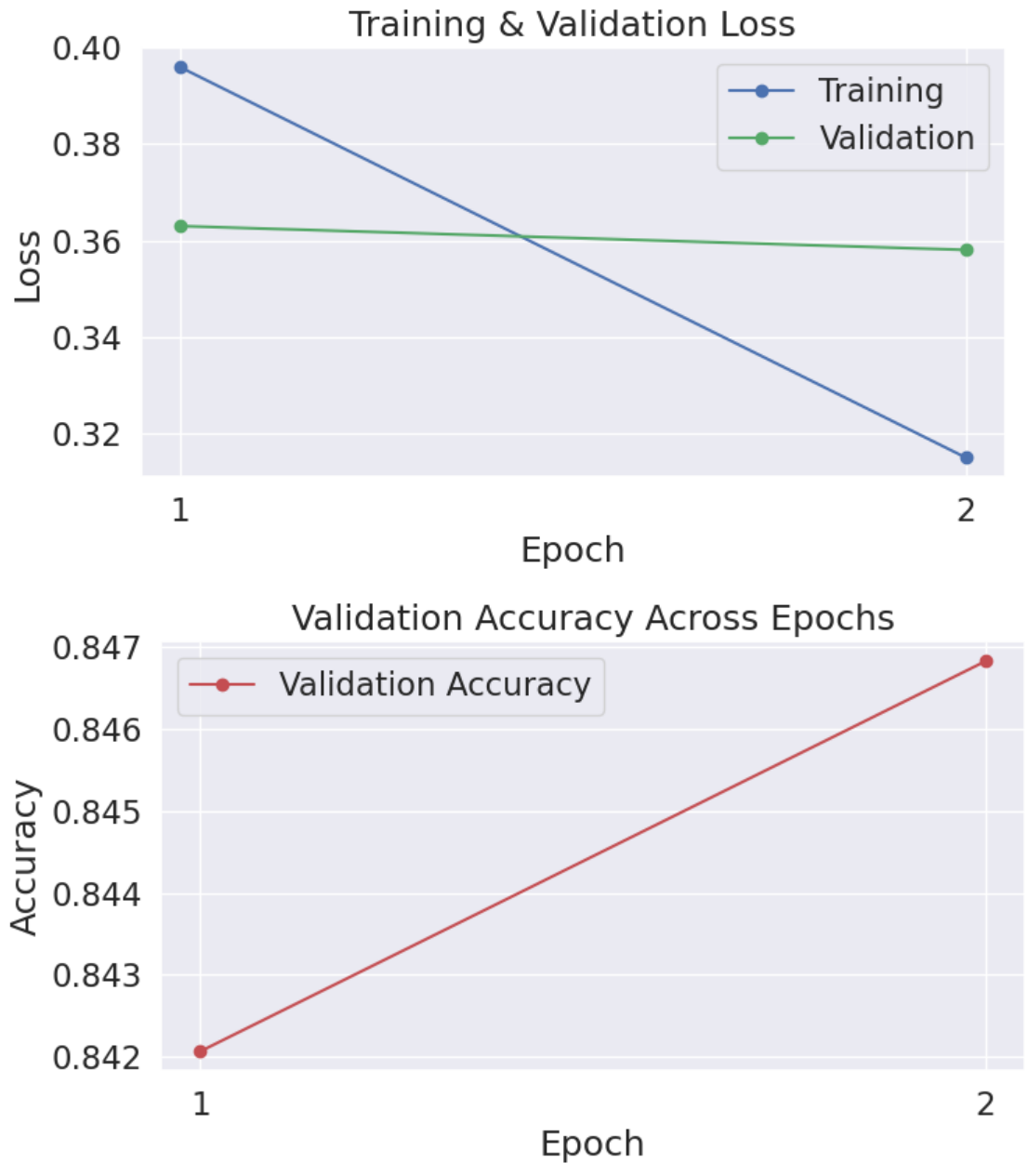


Figure 11: Experiment 3

- Time needed: `0:31:09 (h:mm:ss)`
- This experiment demonstrates no indications of overfitting and achieves an accuracy of 0.8468.

4. Fourth Experiment - Experiment 3 with smaller batch size

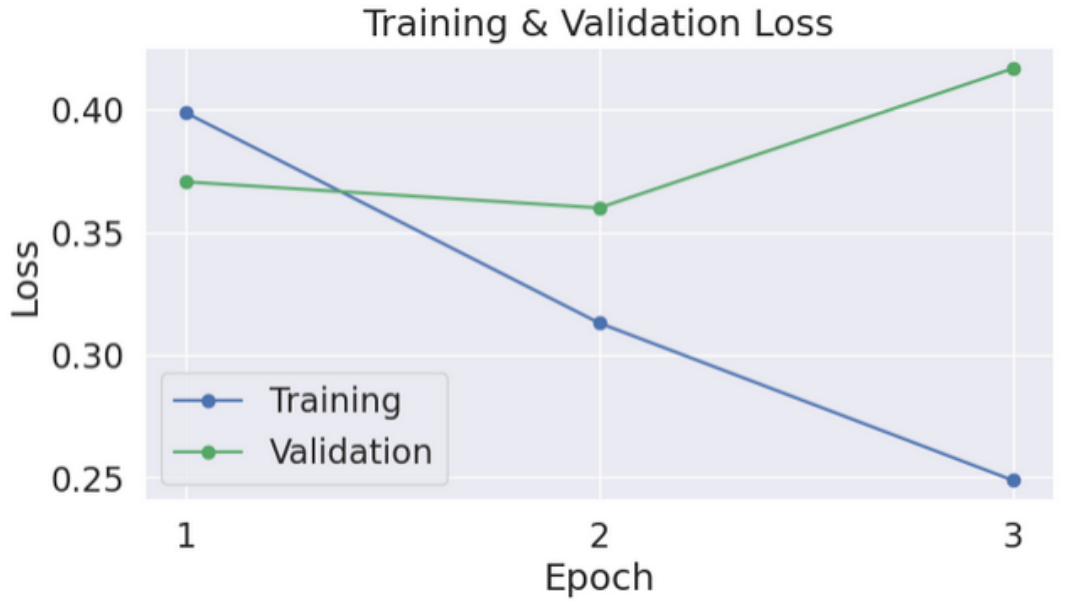
    - This experiment's purpose is to understand if we can have any improvement using a smaller batch size.
    - It uses `batch_size = 16`.
    - It also increased **epochs** to three, since a smaller batch size usually reduces overfitting and this can potentially score better results in later epochs.
    - Configuration:
        - `Batch size: 16`
        - `Number of epochs: 3`
        - `Learning rate: 2e-5`
        - `eps = 1e-8`
        - `num_warmup_steps = 0`
        - `hidden_dropout: 0.1`
        - `attention_dropout: 0.2`
        - `weight_decay: 0.2`
    - Results:

```
Validation Accuracy: 0.8467

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.85     21197
           1       0.85      0.84      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```



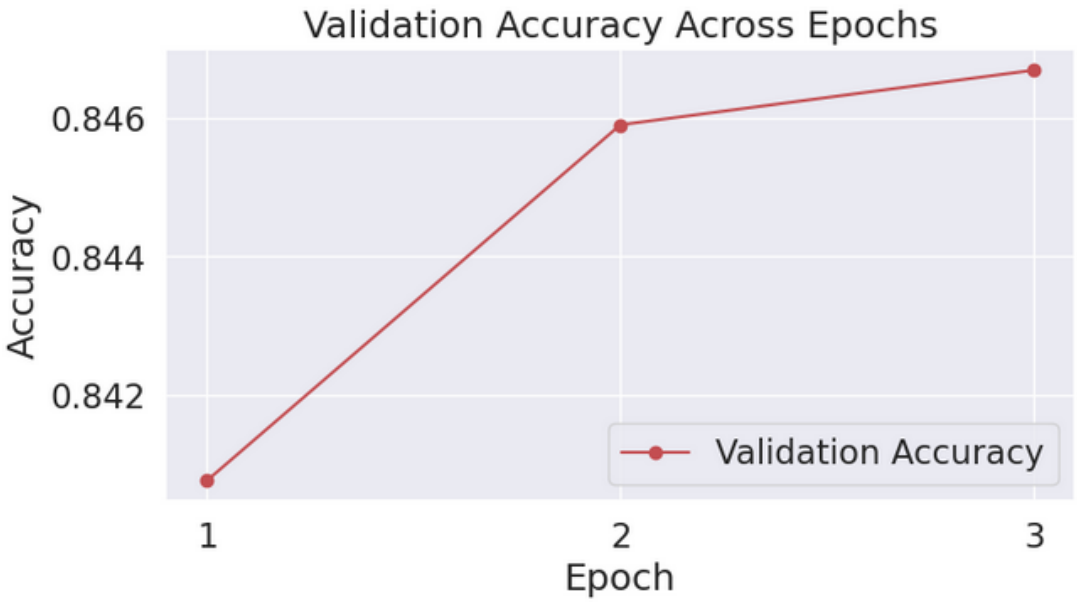


Figure 12: Experiment 4

- Time needed: `0:49:44 (h:mm:ss)`
- The results indicate that there is no real improvement with a smaller batch size and epoch of 3.

- epoch of 2, indicates some better results, but still worse overall performance than Experiment 3.
- For the above reasons **Experiment 3** is considered the best so far.

5. Fifth Experiment - Try warm-up steps

   - This experiment's purpose is to understand if adding some warm-up steps can increase the model's performance.
   - Bert models are pretrained, so jumping in with full learning rate right away can destabilize fine-tuning. A small warm-up (e.g., 0–10% of total steps) helps make training smoother and more effective. This works by linearly increasing the learning rate from 0 to the target learning rate for the first `num_warmup_steps`.
   - Configuration:
     - `Batch size: 32`
     - `Number of epochs: 2`
     - `Learning rate: 2e-5`
     - `eps = 1e-8`
     - `total_steps = len(train_dataloader) * epochs`
     - `num_warmup_steps = int(0.05 * total_steps)`
     - `hidden_dropout: 0.1`
     - `attention_dropout: 0.2`
     - `weight_decay: 0.2`
   - Results:

```
Validation Accuracy: 0.8471

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.85     21197
           1       0.85      0.84      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

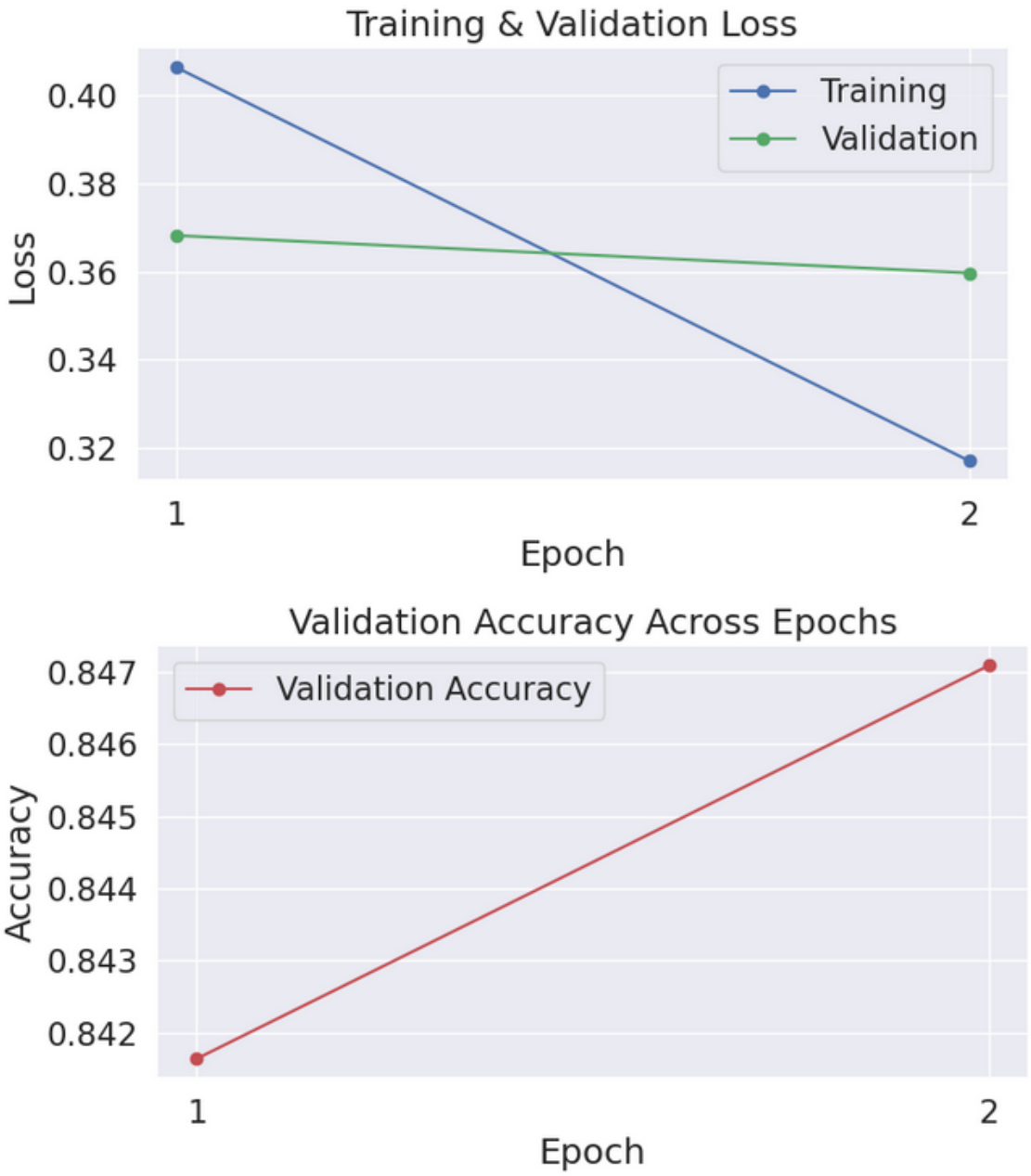


Figure 13: Experiment 5

- Time needed: `0:31:14`
- The overall results are worse than those of Experiment 3. Although the ac-

curacy is slightly higher, the validation loss and classification report metrics are lower.

- For the above reason, **Experiment 3** is considered the best one.

3.3. Hyper-parameter tuning

3.3.1. BERT.

- The final configuration is that of Experiment 8, more precisely:
  - `Batch size: 32`
  - `Number of epochs: 2`
  - `Learning rate: 3e-5`
  - `eps = 1e-8`
  - `num_warmup_steps = 0`
  - `hidden_dropout: 0.1`
  - `attention_dropout: 0.3`
  - `weight_decay: 0.3`
- The BERT model achieves a validation accuracy of 0.8535, indicating strong overall performance. The classification report metrics are all balanced at 0.85–0.86 for both classes, suggesting the model is equally effective at detecting both positive and negative sentiments. Additionally, the macro and weighted averages align closely with the overall accuracy, showing no significant class imbalance or bias in prediction.

```
Validation Accuracy: 0.8535

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.86      0.85     21197
           1       0.86      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

Figure 14: BERT - Classification Report

- The training loss decreases rapidly between epochs 1 and 2, while the validation loss slightly increases, indicating that the model is starting to overfit. Despite this, the validation accuracy still improves slightly, suggesting that the model continues to generalize reasonably well for this epoch. However, the widening gap between training and validation loss is a warning that further training in

another epoch could lead to increased overfitting, something that was proven in the previous experiments.

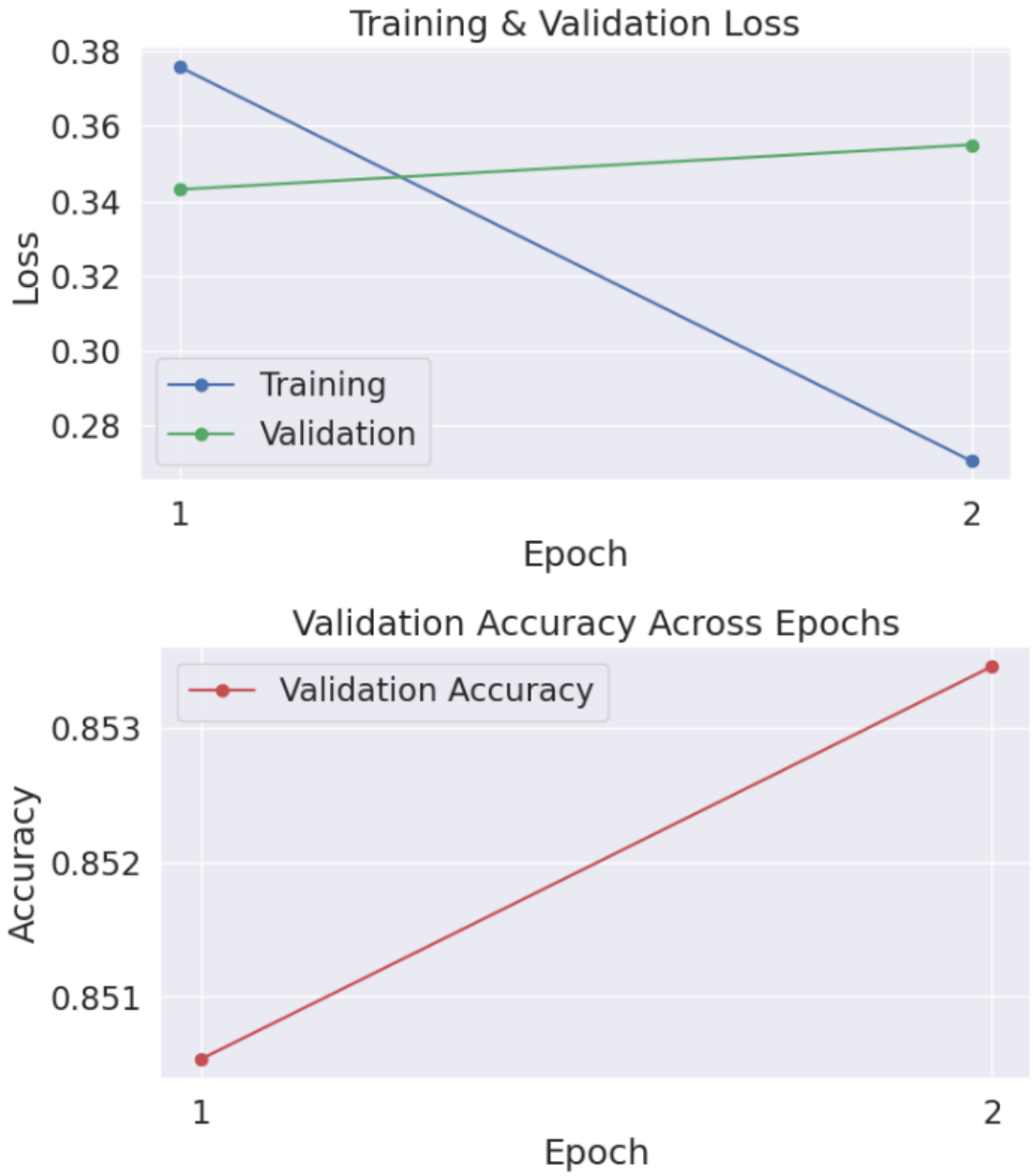


Figure 15: BERT - Learning Curves

3.3.2. DistilBERT.

- The final configuration is that of Experiment 3, more precisely:
  - `Batch size: 32`
  - `Number of epochs: 2`
  - `Learning rate: 2e-5`
  - `eps = 1e-8`

- `total_steps = len(train_dataloader) * epochs`
- `num_warmup_steps = int(0.05 * total_steps)`
- `hidden_dropout: 0.1`
- `attention_dropout: 0.2`
- `weight_decay: 0.2`

- The DistilBERT model achieves a validation accuracy of 0.8468, indicating strong overall performance but slightly lower than the BERT model. The classification report metrics are all at 0.85 for both classes, suggesting the model is equally effective at detecting both positive and negative sentiments. Additionally, just like the BERT model, the macro and weighted averages align closely with the overall accuracy, showing no significant class imbalance or bias in prediction.

```
Validation Accuracy: 0.8468

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85     21197
           1       0.85      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

Figure 16: DistilBERT - Classification Report

- The training loss decreases rapidly between epochs 1 and 2, while the validation loss decreases but a lot slower, indicating that the model could potentially start to overfit in another epoch. Despite this, the validation accuracy still improves slightly, suggesting that the model continues to generalize reasonably well for this epoch. However, the widening gap between training and validation loss is a warning that further training in another epoch could lead to increased overfitting, something that was proven in the previous experiments.

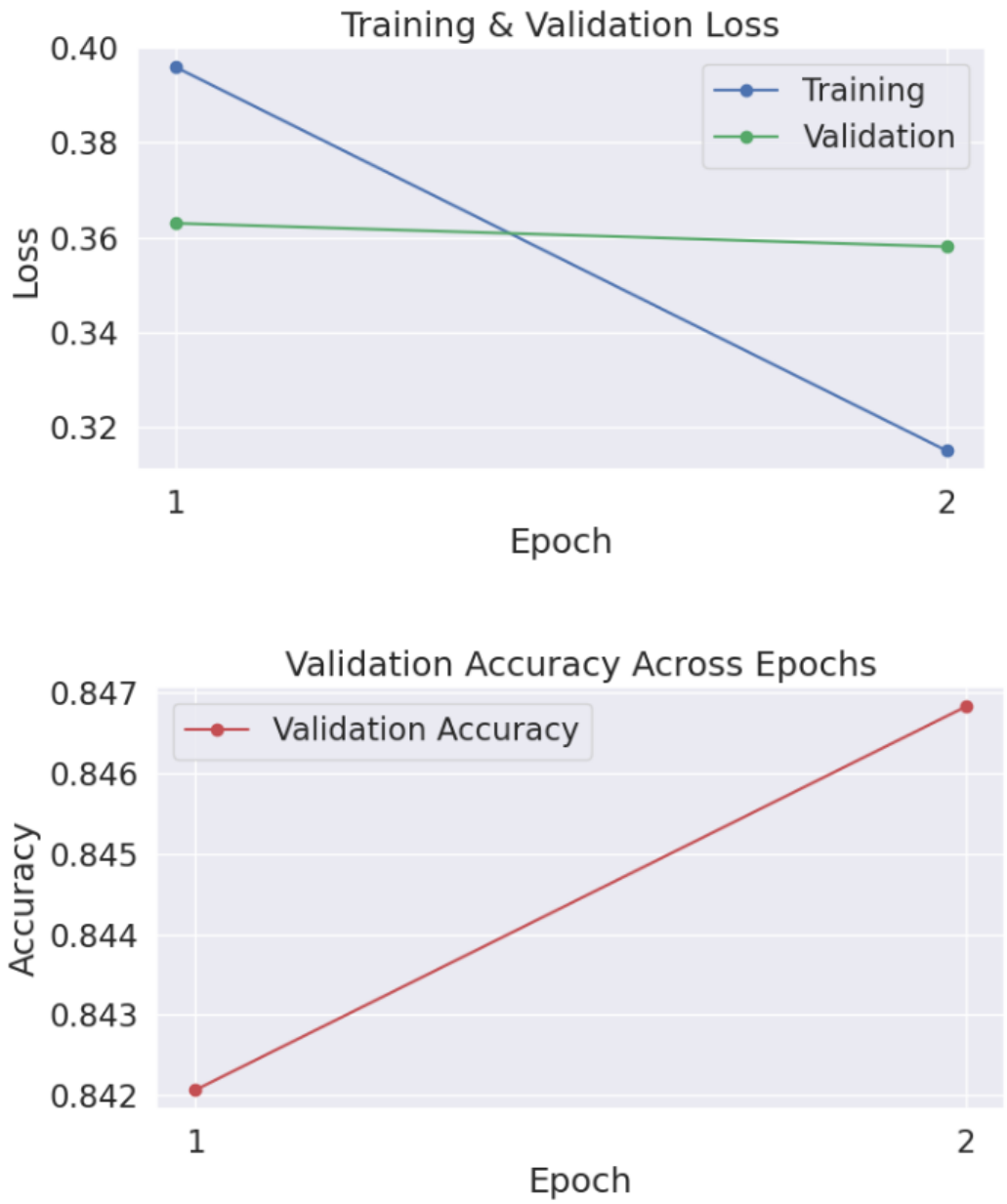


Figure 17: DistilBERT - Learning Curves

3.4. Optimization techniques

1. In both BERT and DistilBERT experiments, Optuna, an optimization framework, was used to run 20 times with different hyperparameters for several epochs to find the best match (lowest val loss).

2. The last Optuna study for both BERT and DistilBERT had the following configuration:

- `training samples: 20% of training data`
- `batch_size: 32 eps: 1e-8`
- `total runs: 20`
- `learning_rate: 5e-5, 3e-5, 2e-5`
- `epochs: 2, 3`
- `hidden_dropout: 0.1, 0.2, 0.3`
- `attention_dropout: 0.1, 0.2, 0.3`
- `weight_decay: 0.0, 0.1, 0.2, 0.3`

3. Results:

- BERT:

	Trial	Validation Loss	params_attention_dropout	params_epoch	params_hidden_dropout	params_learning_rate	params_weight_decay
0	0	0.418070	0.2	3	0.3	0.00005	0.0
1	1	0.503414	0.3	3	0.1	0.00005	0.2
2	2	0.393722	0.1	2	0.1	0.00002	0.2
3	3	0.390611	0.3	2	0.1	0.00003	0.3
4	4	0.400425	0.2	2	0.3	0.00005	0.2
5	5	0.447955	0.3	3	0.1	0.00003	0.2
6	6	0.418584	0.2	3	0.2	0.00002	0.3
7	7	0.425632	0.3	3	0.3	0.00005	0.2
8	8	0.430293	0.2	3	0.3	0.00005	0.3
9	9	0.469454	0.1	3	0.1	0.00002	0.0
10	10	0.398464	0.2	2	0.3	0.00002	0.1
11	11	0.391243	0.2	2	0.1	0.00002	0.2
12	12	0.401370	0.1	2	0.2	0.00005	0.1
13	13	0.587362	0.1	3	0.1	0.00005	0.1
14	14	0.401583	0.2	3	0.3	0.00002	0.0
15	15	0.444725	0.3	3	0.1	0.00003	0.2
16	16	0.438241	0.1	3	0.3	0.00005	0.1
17	17	0.481473	0.2	3	0.2	0.00005	0.1
18	18	0.421897	0.3	3	0.2	0.00003	0.3
19	19	0.413096	0.3	3	0.3	0.00002	0.0

Figure 18: BERT - Optuna Study Results

- DistilBERT:

	Trial	Validation Loss	params_attention_dropout	params_epoch	params_hidden_dropout	params_learning_rate	params_weight_decay
0	0	0.433325	0.2	3	0.3	0.00005	0.0
1	1	0.490467	0.3	3	0.1	0.00005	0.2
2	2	0.398665	0.1	2	0.1	0.00002	0.2
3	3	0.402213	0.3	2	0.1	0.00003	0.3
4	4	0.405834	0.2	2	0.3	0.00005	0.2
5	5	0.428273	0.3	3	0.1	0.00003	0.2
6	6	0.414444	0.2	3	0.2	0.00002	0.3
7	7	0.429372	0.3	3	0.3	0.00005	0.2
8	8	0.431017	0.2	3	0.3	0.00005	0.3
9	9	0.438741	0.1	3	0.1	0.00002	0.0
10	10	0.406745	0.2	2	0.3	0.00002	0.1
11	11	0.397118	0.2	2	0.1	0.00002	0.2
12	12	0.415624	0.1	2	0.2	0.00005	0.1
13	13	0.542146	0.1	3	0.1	0.00005	0.1
14	14	0.405825	0.2	3	0.3	0.00002	0.0
15	15	0.427532	0.3	3	0.1	0.00003	0.2
16	16	0.441479	0.1	3	0.3	0.00005	0.1
17	17	0.452253	0.2	3	0.2	0.00005	0.1
18	18	0.416736	0.3	3	0.2	0.00003	0.3
19	19	0.414508	0.3	3	0.3	0.00002	0.0

Figure 19: DistilBERT - Optuna Study Results

3.5. Evaluation

- I evaluated the predictions using accuracy, classification report metrics, and Plots.
- For the BERT model, all the metrics reached around 85%, while for the DistilBERT 84-85%, showing the general stability and good performance of the model. These results, and especially the F1-score, which can be described as the harmonic mean of the precision and recall of a classification model, reveal a good performance in recognizing positive cases while minimizing false positives and false negatives.
- Learning curves were plotted to analyze model performance across different experiments, helping to detect overfitting or underfitting trends.
- The following table shows the average metrics between the two classes of each experiment.
- BERT Classification Report:

```
Validation Accuracy: 0.8535

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.86      0.85     21197
           1       0.86      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

Figure 20: BERT - Classification Report

- DistilBERT Classification Report:

```
Validation Accuracy: 0.8468

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85     21197
           1       0.85      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

Figure 21: DistilBERT - Classification Report

3.5.1. ROC curve.

- The curves for both the BERT and the DistilBERT model are almost identical. For this reason, the below comments apply to both of them.
- The ROC curves demonstrate strong performance, with the classifiers achieving a high True Positive Rate (TPR) even at low False Positive Rates (FPR), significantly outperforming the random classifier (red dashed line).
- An overall AUC (Area Under the ROC Curve) of 0.93 indicates that the models have good discrimination ability between positive and negative tweets, correctly ranking a randomly chosen positive example above a negative one roughly 93% of the time.

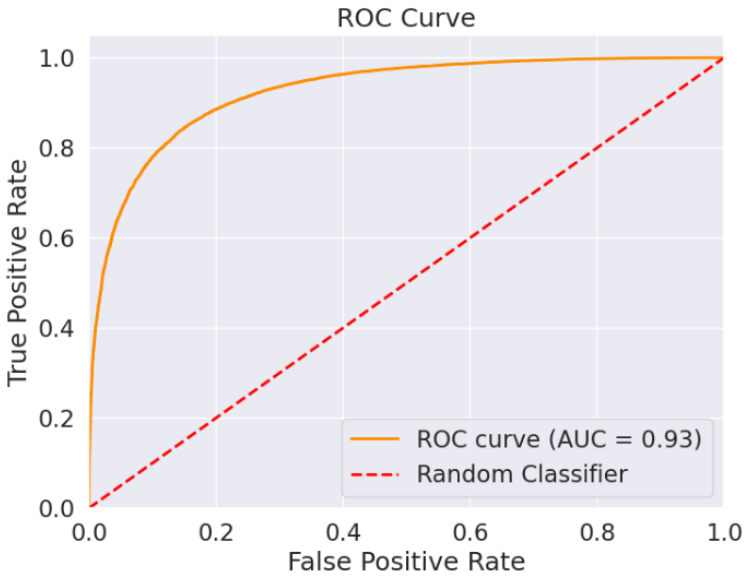


Figure 22: BERT - ROC Curve

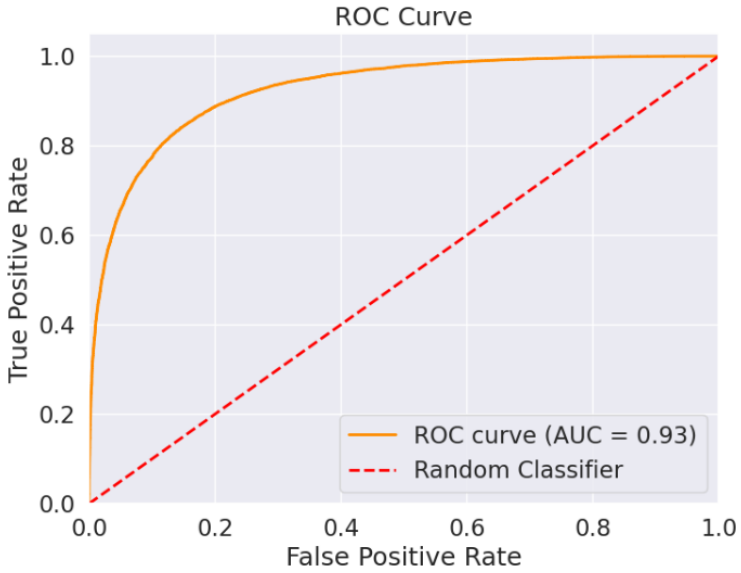


Figure 23: DistilBERT - ROC Curve

3.5.2. Learning Curve.

- For the BERT model, the training loss decreases significantly between epochs, while the validation loss slightly increases, suggesting the beginning of overfitting. However, the validation accuracy continues to improve, indicating that the model still generalizes reasonably well at this stage. These curves suggest that further training in more epochs could lead to stronger overfitting. For the above reason, the current number of epochs provides a good balance between learning and generalization.

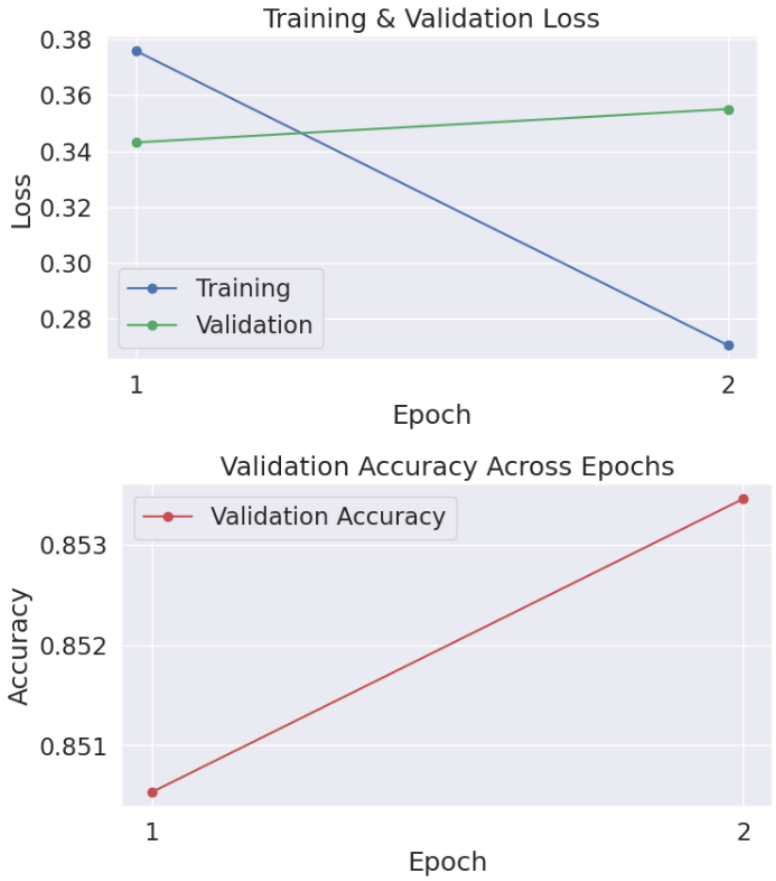


Figure 24: BERT - Learning Curves

- For the DistilBERT model, the training loss decreases significantly between epochs, while the validation loss decreases but a lot slower, indicating that the model could potentially start to overfit in another epoch. However, the validation accuracy continues to improve, indicating that the model still generalizes reasonably well at this stage. These curves suggest that further training in more epochs could lead to stronger overfitting. For the above reason, the current number of epochs provides a good balance between learning and generalization.

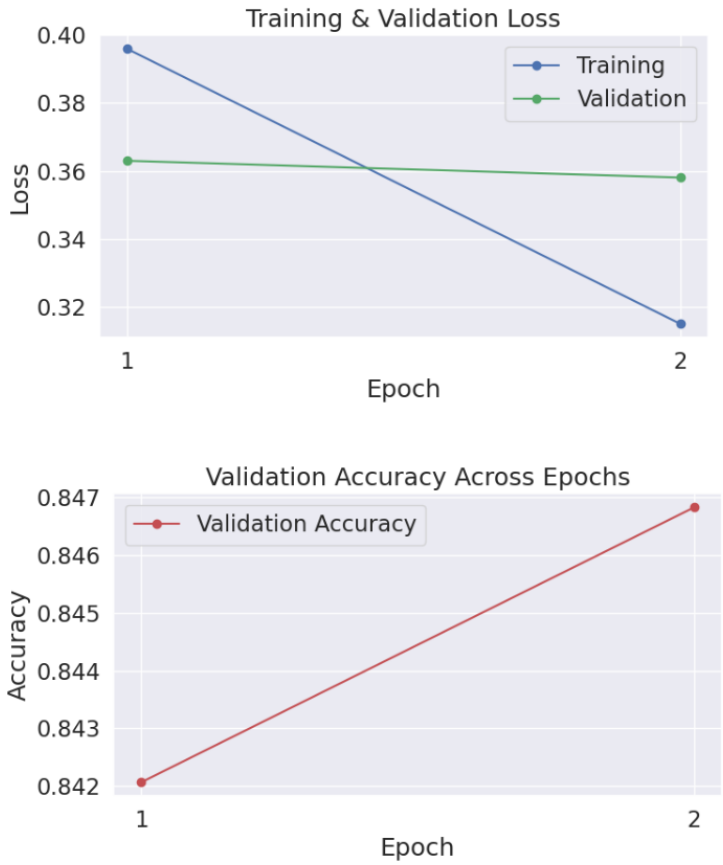


Figure 25: DistilBERT - Learning Curves

3.5.3. Confusion matrix.

- The model performs well overall, as the numbers for correct predictions (TP & TN) are significantly higher than incorrect ones.
- For the BERT model:
  - True Negatives (TN): 18006
  - False Positives (FP): 3191
  - False Negatives (FN): 3293
  - True Positives (TP): 17906

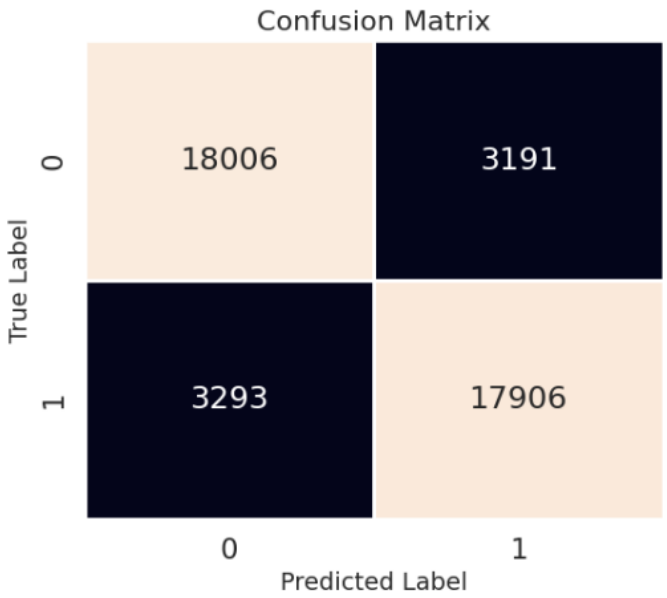


Figure 26: BERT - Confusion Matrix

- For the DistilBERT model:
  - True Negatives (TN): 17979
  - False Positives (FP): 3218
  - False Negatives (FN): 3276
  - True Positives (TP): 17923

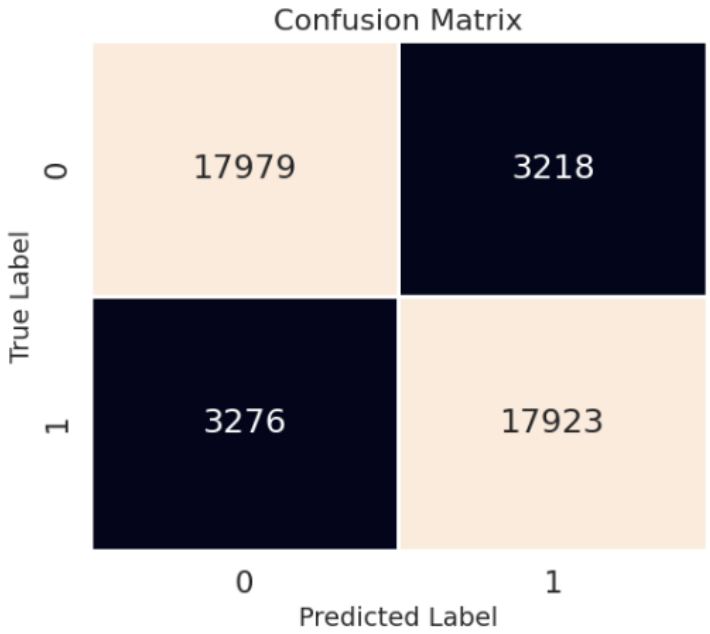


Figure 27: DistilBERT - Confusion Matrix

4. Results and Overall Analysis

4.1. Results Analysis

- My final results show an accuracy of around 85.5% for the BERT model and 84.5% for the DistilBERT model. Metrics such as precision, recall, and f1-score, alongside the Learning Curves plot, also show that the models are balanced.
- More experiments I would make would be to run an Optuna optimization for every experiment of the independent experiments (e.g., reducing batch size), but this was not an available option for me due to time limitations.

4.1.1. Best trial.

- Results of best trial (BERT):

```
Validation Accuracy: 0.8535

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.86      0.85     21197
           1       0.86      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

Figure 28: BERT - Best Trial - Classification Report

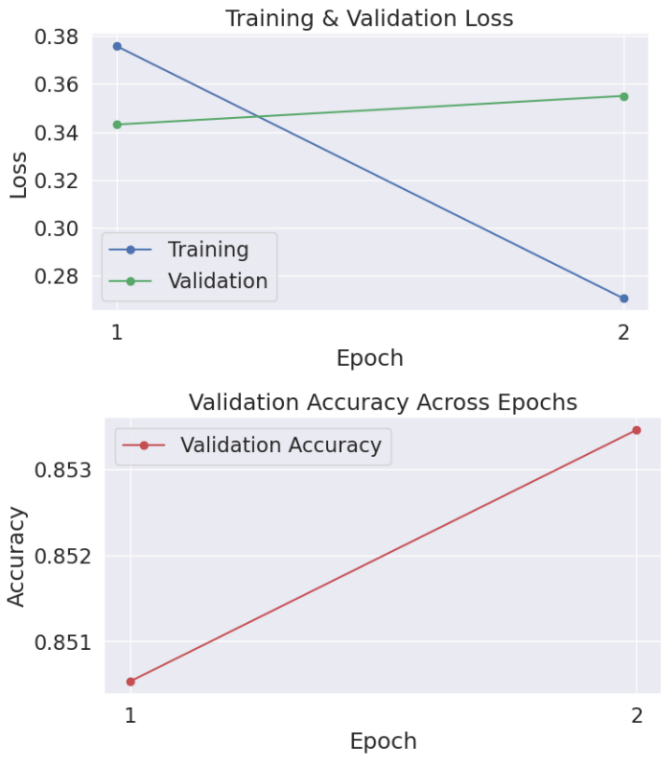


Figure 29: BERT - Best Trial - Learning Curves

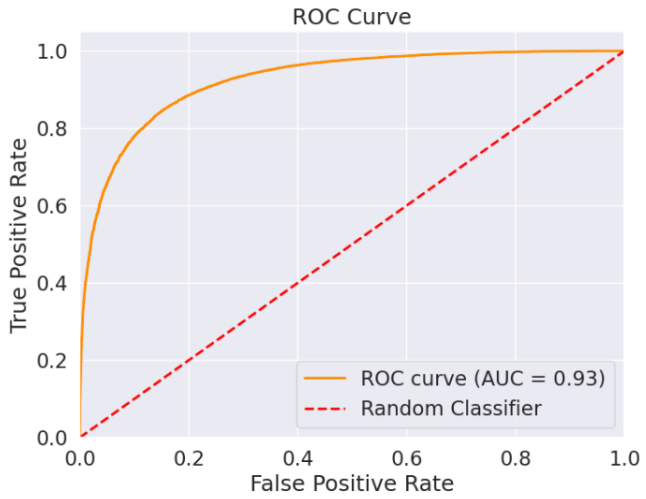


Figure 30: BERT - Best Trial - ROC Curve

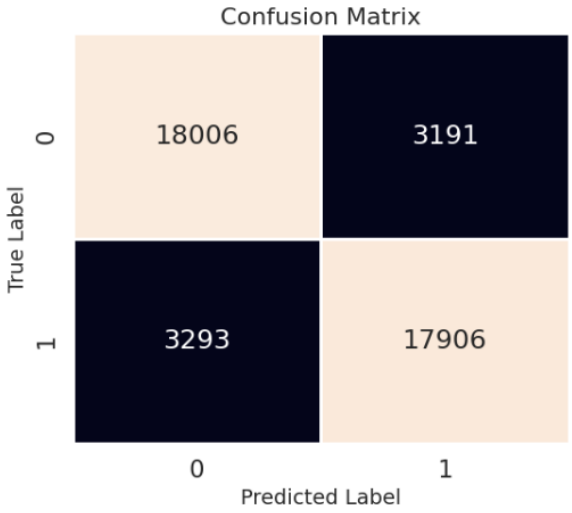


Figure 31: BERT - Best Trial - Confusion Matrix

- Results of best trial (DistilBERT):

```
Validation Accuracy: 0.8468

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85     21197
           1       0.85      0.85      0.85     21199

    accuracy                           0.85     42396
   macro avg       0.85      0.85      0.85     42396
weighted avg       0.85      0.85      0.85     42396
```

Figure 32: DistilBERT - Best Trial - Classification Report

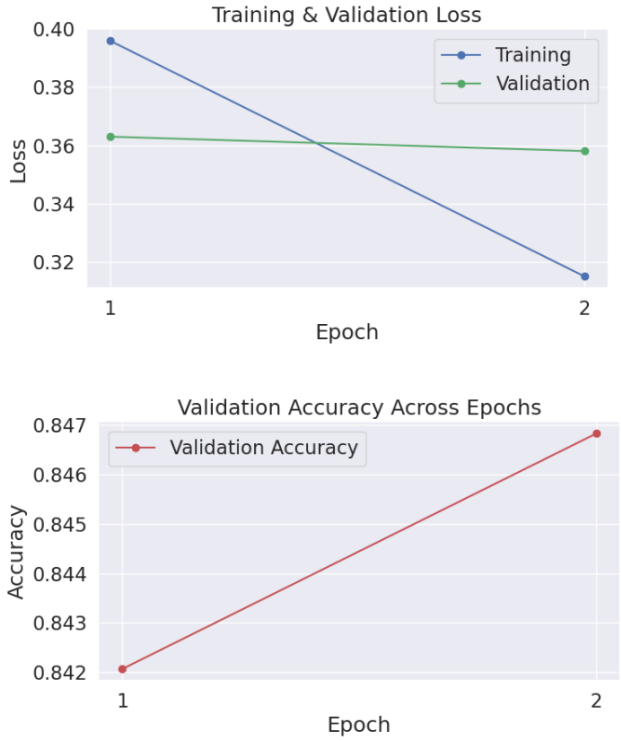


Figure 33: DistilBERT - Best Trial - Learning Curves

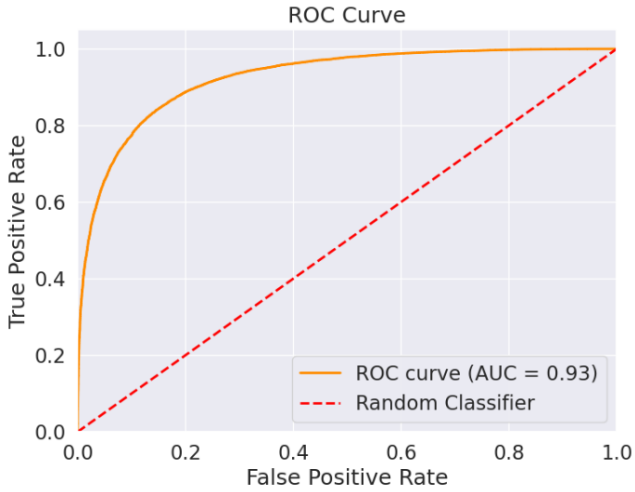


Figure 34: DistilBERT - Best Trial - ROC Curve

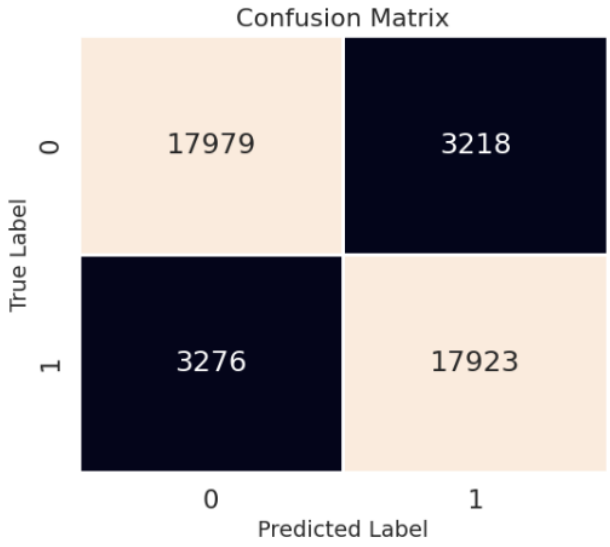


Figure 35: DistilBERT - Best Trial - Confusion Matrix

- More information about those trials in the previous sectors of the report.

4.2. Comparison with the first project

- While the logistic regression baseline in Project 1 performed surprisingly well (around 80% accuracy), Project 3 models clearly outperform it with 85%+ accuracy and stronger F1-scores.
- Project 1 relied on TF–IDF features, which are sparse and non-contextual, while Project 3 uses dense, contextualized representations from pretrained language models.
- The difference in model complexity is of high importance. Logistic regression is much simpler than the BERT models, which are deep, nonlinear, and pretrained on massive corpora, giving them a major head start on language understanding.

4.3. Comparison with the second project

- The transformer-based models used in Project 3 (BERT and DistilBERT) significantly outperform the neural network from Project 2, achieving validation accuracies of 84-85% compared to 78.6% in the previous assignment.
- Unlike the static Word2Vec embeddings used in Project 2, the models in Project 3 leverage contextual embeddings, allowing them to better capture meaning and word dependencies within tweets.

- The fine-tuning of pretrained transformers also makes them more robust and data-efficient, reducing the need for extensive architecture tuning compared to the manually designed network in Project 2.
- However, this performance gain comes with a computational cost, training BERT requires substantially more time and resources than the simpler two-layer network.