Final Project Report-Author Classification Using BERT

Table of Contents

Course: SE4475
Student: Fahrettin Ege Bilge

Submission Date: 23/01/2025

1. Overview

This project addresses a **multi-class text classification problem** to identify authors of Turkish newspaper articles from a dataset of 30 distinct authors. The solution leverages a **fine-tuned BERT model** and **5-fold cross-validation** to ensure robustness and generalizability.

---

2. Methodology

2.1 Model Selection

- **Pre-trained BERT Model**: The `dbmdz/bert-base-turkish-cased` model was chosen due to its pre-training on Turkish text, enabling effective capture of language-specific syntax and semantics.
- **Fine-tuning**: The model was adapted for sequence classification by adding a classification head with 30 output neurons (one per author).

2.2 Training Strategy

- **Dynamic Learning Rate**:
  - **Cosine Decay with Warmup**: Initial warmup stabilizes training, followed by cosine decay for smooth convergence.
  - **Initial LR**: `3e-5`, **Minimum LR**: `1e-5`.
- **Regularization**:
  - **Weight Decay**: Linearly adjusted from `0.01` to `0.001` to balance model complexity.
  - **Label Smoothing**: Applied to **CrossEntropyLoss** (`smoothing=0.1`) to mitigate overconfidence in predictions.
- **5-Fold Cross-Validation**: Ensures unbiased evaluation by training and validating on different subsets of the data, reducing overfitting risks.

2.3 Optimization

- **Early Stopping**: Halts training if validation F1-score does not improve for 3 epochs, preventing overfitting.
- **Gradient Clipping**: Limits gradient norms to `1.0` for stable updates.

2.4 Evaluation Metrics

- **Primary Metrics**: Macro-averaged Precision, Recall, and F1-score (chosen due to class balance).
- **Confusion Matrices**: Visualized per-fold performance to identify class-specific strengths/weaknesses.

---

3. Implementation Details

3.1 Data Preparation

- **Dataset**: 1,500 articles (50 per author) loaded into `TextDataset`, tokenized with a max sequence length of 512 (BERT's limit).
- **Stratified Sampling**: Preserves class distribution across folds.

3.2 Training Configuration

- **Batch Size**: 4 (due to GPU memory constraints).
- **Epochs**: 10 (with early stopping).
- **Hardware**: Utilized Apple MPS (Metal Performance Shaders) for accelerated training.

---

4. Results

4.1 Performance Metrics (Average Across Folds)

Metric	Score
F1-Score	91.76%
Precision	92.74%
Recall	91.93%

- **Per-Class Consistency**: Most classes achieved F1-scores >90%, with minor variations (e.g., Class 3 at `72.02%` due to harder distinctions).

4.2 Per-Class Performance

Overall Performance Metrics

		Precision	Recall	F-Score
0	1	0.911111	0.960000	0.932057
1	2	0.906667	0.920000	0.911292
2	3	0.856429	0.640000	0.720224
3	4	0.981818	0.960000	0.969424
4	5	0.960000	0.880000	0.916725
5	6	0.981818	1.000000	0.990476
6	7	1.000000	1.000000	1.000000
7	8	1.000000	1.000000	1.000000
8	9	0.905000	0.920000	0.901098
9	10	0.852525	0.780000	0.804637
10	11	0.930303	1.000000	0.962771
11	12	0.896970	1.000000	0.944589
12	13	0.963636	1.000000	0.980952
13	14	1.000000	1.000000	1.000000
14	15	0.766480	0.940000	0.840207
15	16	0.891818	0.920000	0.903586
16	17	0.963636	0.980000	0.970426
17	18	1.000000	0.720000	0.831957
18	19	0.918462	0.940000	0.925995
19	20	0.892857	0.920000	0.897554
20	21	0.924444	0.760000	0.827601
21	22	1.000000	0.860000	0.923977
22	23	0.830000	0.860000	0.843636
23	24	0.981818	1.000000	0.990476
24	25	0.948485	0.980000	0.961768
25	26	0.981818	1.000000	0.990476
26	27	0.832634	1.000000	0.905342
27	28	0.893442	0.940000	0.912810
28	29	0.895556	0.820000	0.854620
29	30	0.955556	0.880000	0.914620
30	Average	0.927443	0.919333	0.917643

Figure 1: Overall Performance Metrics

4.3 Key Visualizations

- **Loss Curves**: Training/validation loss convergence confirmed effective learning.
- **Confusion Matrices**: Highlighted high diagonal accuracy, with occasional misclassifications between stylistically similar authors.

4.3.1 Sample Folds

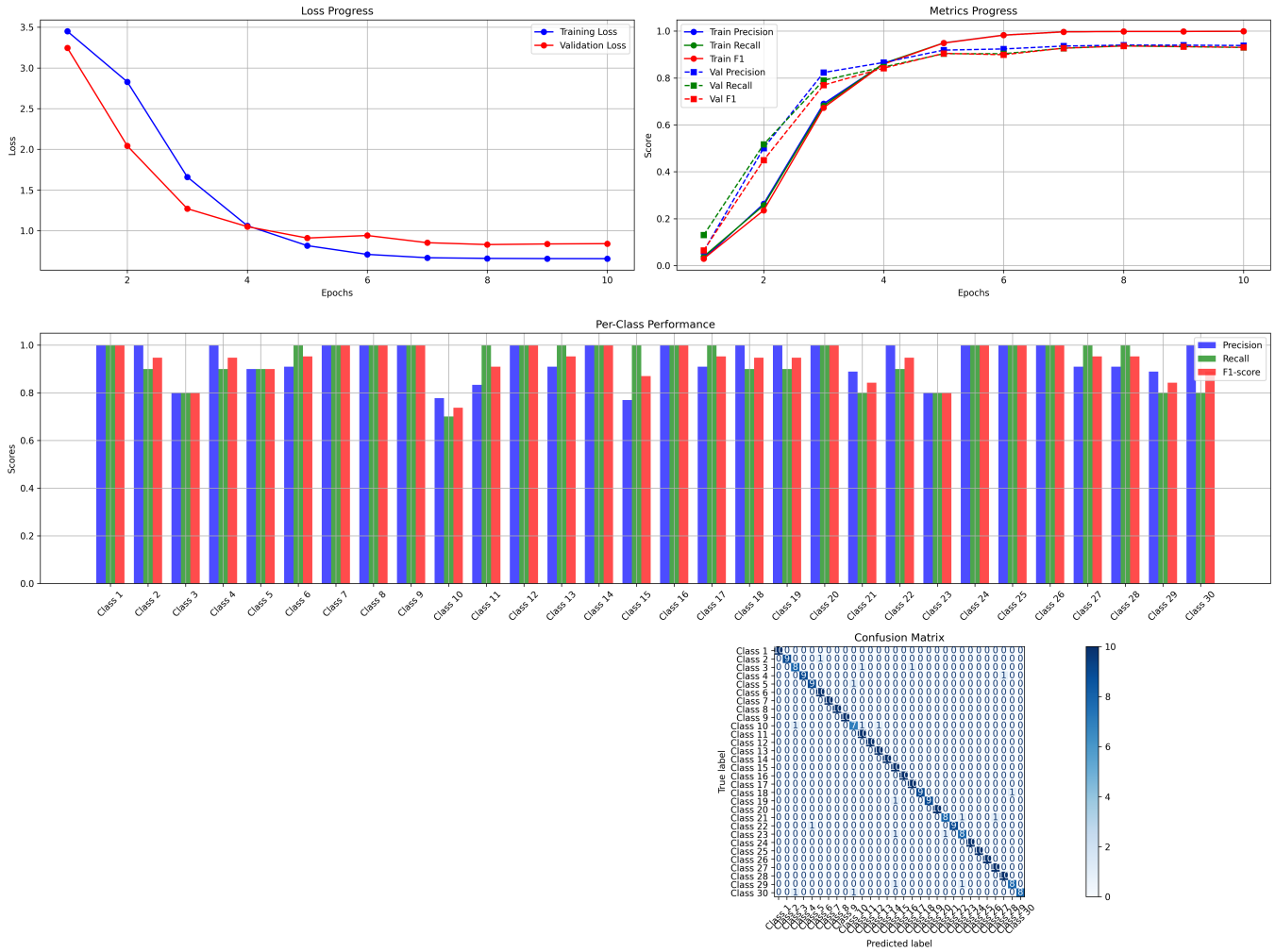

Figure 2: Fold-1 Plots

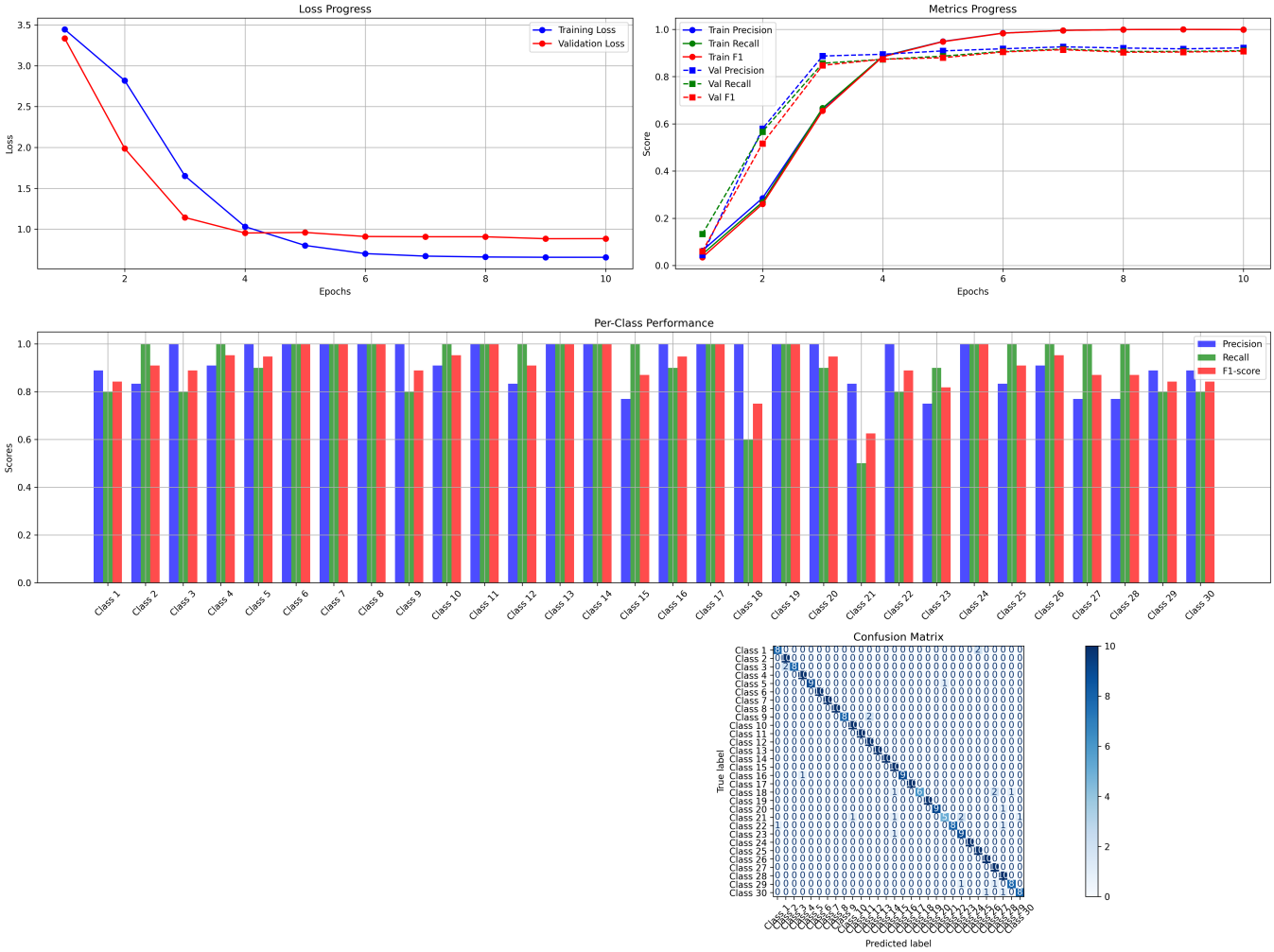

Figure 3: Fold-3 Plots

---

5. Conclusion

5.1 Strengths

- **Language-Specific Pretraining**: The Turkish BERT model provided strong baseline performance.
- **Robust Validation**: Cross-validation ensured reliable metrics.

5.2 Limitations & Future Work

- **Class Imbalance**: While balanced, some authors' writing styles may overlap, requiring data augmentation.
- **Model Variants**: Testing larger models (e.g., RoBERTa) could improve accuracy.

---

6. Deliverables

1. **Source Code**: Jupyter notebook (`identification.ipynb`) with training logic.
2. **Performance Reports**:
   - Fold-wise CSVs (`fold_metrics/`).
   - Confusion matrices and loss curves (`plots/`).
3. **Configuration**: Hyperparameters in the notebook.

Submitted by: Fahrettin Ege Bilge
ID: 21070001052