# Project 1: Hyperparameter Tuning

**by Nils Schell (nils.schell@stud.hslu.ch)**

This report outlines the hyperparameter optimization of a distilbert-base-uncased model for the MRPC paraphrase detection task. Through a combined approach of manual trial and error and automated trials using the Optuna tool, the model achieved an **optimal F1 score of ~0.9019**. The main finding is that, although the final performance improvement over the baseline was minimal due to the model's high robustness, the automated approach proved to be slightly more efficient than the manual process.

The structure of the project required that the number of epochs (3) could not be changed. This short epoch limit requires highly efficient training, making hyperparameter configuration crucial for maximizing performance before overfitting occurs. To enable persistent automatic logging of metrics, parameters, and model artefacts, MLflow (version: 3.4.0) was used in conjunction with PyTorch Lightning. This environment ensured that the full configuration and result of every run were recorded and comparable.

A list of seven hyperparameters, which is shown in the table on the right, was investigated in 15 try and error manual training runs. The process began with testing extreme combinations of the four main hyperparameters to quickly establish performance boundaries. The remaining parameters max_seq_length, beta1 and gradient_clipping_val were each tested once to check their influence. The top result of the 15 trials so far had a F1 score of ~0.8998.

| Nr. | Hyperparameter | Type |
|---|---|---|
| 1 | learning_rate | Float |
| 2 | weight_decy | Float |
| 3 | train_batch_size | Int |
| 4 | max_seq_length | Int |
| 5 | warmup_ratio | Float |
| 6 | beta1 | Float |
| 7 | gradient_clipping_val | Float |

*Figure 1: used Hyperparameters*

After that, the aim was to narrow down this set of hyperparameters to the three most promising ones for further tuning. The primary analytical tool used was the Parallel Coordinates Plot (PCP), which visualises all hyperparameter axes against the F1 score. The PCP chart revealed high sensitivity in the learning_rate, train_batch_size, and weight_decay axes. These three were selected for Week 2 fine-tuning.

- Learning Rate (LR): The LR axis shows bimodal sensitivity, with peak performance occurring at $2e^{-5}$ and $5e^{-5}$, while intermediate values produced poorer results. This fluctuation confirms that LR is critical for fine-tuning.
- Train Batch Size (BS): Best-performing runs were heavily concentrated in the upper tested range (32 to 64). This clustering indicated the model benefits from larger effective batch sizes.
- Weight Decay (WD): WD also exhibited a bimodal optimum as LR, with high scores at both 0.0 and 0.1 and performance loss in between. This sensitivity to regularization strength makes precise tuning essential.

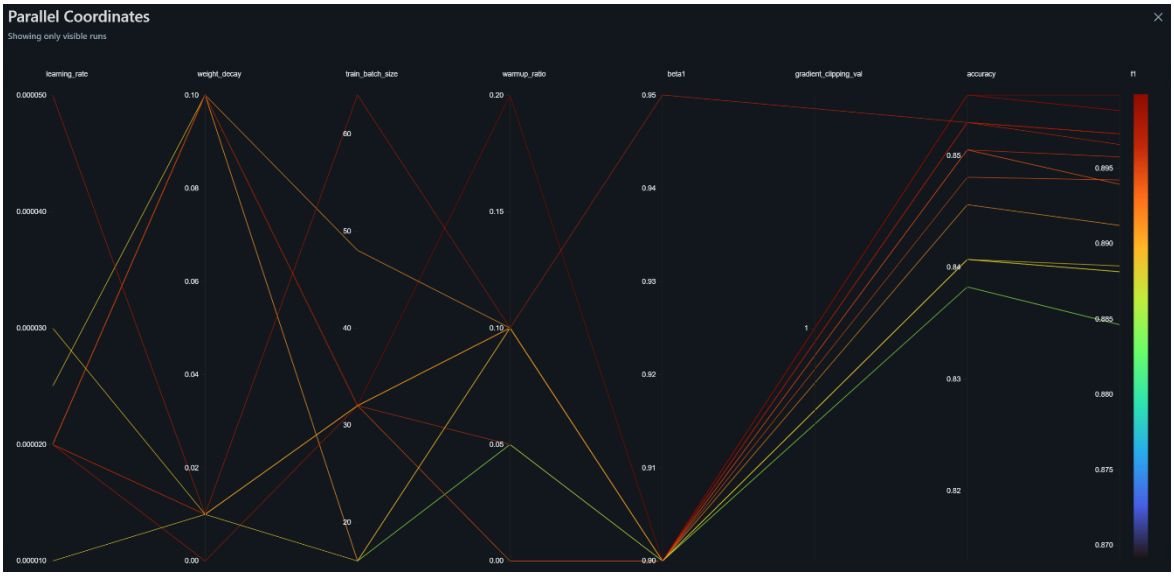


*Figure 2: Screenshot of MLflow Parallel Coordinates Plot*

After selecting the three most promising hyperparameters, a systematic grid search of 18 runs (3 × 2 × 3 = 18) was performed manually. The values of the constant parameters and the ranges of the variable parameters are shown in the table on the right. This manual exploration yielded to the best manual setting, achieving a peak F1 score of ~0.8991 which is statistically slightly better than the trying out phase. The optimal configuration is:
**LR = 3.5e$^{-5}$, BS = 64 and WD = 0.1**

| | |
|---|---|
| Constant Parameters | beta1 = 0.9 \| gradient_clipping_val = 1.0 \| max_seq_length = 128 \| warmup_ratio = 0.1 |
| Variable Parameters | learning_rate = $[2e^{-5}, 5e^{-5}]$<br>train_batch_size = $[32, 64]$<br>weight_decay = $[0.0, 0.1]$ |

*Figure 3: Hyperparameter Selection*

| | | |
|---|---|---|
| Trying Out | accuracy ~= 0.8554 | f1 ~= 0.8998 |
| Systematic Manuel Search | accuracy ~= 0.8554 | f1 ~= 0.8991 |

*Figure 4: Manual Search Results*

The nearly identical top results from runs with quite different hyperparameters demonstrate the robustness of the model and the interdependence of the parameters. This is known as compensatory performance, whereby the model found different paths to the same high result. As the objective is to identify the optimal setting, this indicates a highly sensitive area of the search space.

For Week 3, the Optuna library with the TPE (Tree-structured Parzen Estimator) sampler was used to automatically search for the optimal hyperparameters across 18 trials. Therefore, still the same three chosen hyperparameters LR, BS and WD were used. The automated search successfully found a slightly superior result, demonstrating the benefit of Bayesian sampling. The best Optuna setting reached a peak F1 score of ~0.9019, which was ~0.0027 points higher than the best manual score. This confirms the efficiency of the automated approach within the tight run budget. The best setting was found during the tenth trial, after which the automated optimisation didn't improve. The values for LR, BS and WD for the best run are below in the final configuration for reproducibility section.

| Run Name | Created | Da | Duration | Source | Model | accuracy | f1 |
|---|---|---|---|---|---|---|---|
| MRPC_Optuna_Trial_10 | 15 minutes ago | - | 1.6min | C:\Users\... | - | 0.860294103... | 0.901893258... |
| MRPC_Grid_LR=3.5e-05_B... | 1 hour ago | - | 1.6min | C:\Users\... | - | 0.855392158... | 0.899145305... |
| MRPC_Grid_LR=3.5e-05_B... | 1 hour ago | - | 1.6min | C:\Users\... | - | 0.855392158... | 0.899145305... |
| MRPC_Grid_LR=2.0e-05_B... | 1 hour ago | - | 1.8min | C:\Users\... | - | 0.855392158... | 0.898799300... |

*Figure 5: Screenshot of the MLflow Run Table of the best Runs*

**Final Configuration for Reproducibility:**
The absolute highest performance across all 51 runs, manual and automated, was achieved by the Optuna tool with an F1 score of ~0.9019 and an accuracy of ~0.8603. The difference between the F1 score and Accuracy is because F1 provides a more balanced assessment of precision and recall.

| | | |
|---|---|---|
| beta1 = 0.9 | learning_rate = 2.642988043609753e-05 | num_labels = 2 |
| eval_batch_size = 32 | model_name_or_path = distilbert-base-uncased | task_name = mrpc |
| eval_splits = ['validation'] | weight_decay = 0.032716372623540435 | train_batch_size = 64 |
| max_seq_length = 128 | gradient_clipping_val = 1.0 | warmup_ratio = 0.1 |

*Figure 6: Table of Parameters for Final Configuration*

**Conclusion:**
The most interesting finding was the model's high robustness, which resulted in a flat optimisation landscape. This means that minimal gains were found, making the choice between manual and automated tuning less impactful than expected. In retrospect, in the initial Week 1 try and error runs I should have tested a wider, more extreme range of hyperparameters, for example lower learning rates and more extreme weight decay values. Testing these wider extremes would have resulted in clearer separations in the parallel coordinates plot, making the final selection of the top three most promising parameters easier and more decisive. Technically, an unexpected MLflow checkpoint logging error required a manual workaround, which is a critical integration and costs valuable development time. Otherwise, the process was successful. Future iterations would benefit from more allowed epochs and trials to provide a clearer data basis for decisions.