Improving Learning Stability in Sentiment Analysis via Hyperparameter Tuning

Juan Carlos Miguel V. Timoteo
Computer Studies & Engineering
José Rizal University
Mandaluyong City, Philippines
juancarlosmiguel.timoteo@my.jru.edu

***Abstract*— This paper presents a systematic approach to fine-tuning a DistilBERT model for sentiment analysis on a dataset with significant class imbalance. The primary objective is to enhance learning stability and model generalization by experimenting with key hyperparameters, including batch size, weight decay for regularization, and learning rate warmup steps. The methodology involves comprehensive data preprocessing, including the removal of English and Tagalog stopwords, and the implementation of a weighted loss function within a custom trainer to counteract the skewed data distribution. Four distinct experiments were conducted, varying per_device_train_batch_size (8, 16, 32), weight_decay (0.0, 0.01, 0.1), and warmup_steps (0, 200, 500). Model performance was evaluated using accuracy, weighted precision, recall, and F1-score. The results demonstrate that a larger batch size of 32 combined with strong L2 regularization (weight decay of 0.1) and a substantial warmup period (500 steps) yielded the best validation F1-score (0.9945), indicating superior stability and generalization on the imbalanced dataset.**

Keywords—sentiment analysis, class imbalance, natural language processing, DistilBERT, weighted loss, multilingual models, text classification

I. Introduction

Sentiment analysis, a subfield of Natural Language Processing (NLP), is crucial for understanding public opinion, customer feedback, and social media trends. Transformer-based models, such as BERT and its derivatives, have achieved state-of-the-art results on many NLP tasks [1]. However, a common challenge in real-world applications is class imbalance, where one class significantly outnumbers others, leading to models that are biased towards the majority class.

Fine-tuning pre-trained models like distilbert-base-multilingual-cased [2] is an effective technique, but its success is highly dependent on the choice of hyperparameters. Parameters such as batch size, regularization strength, and learning rate scheduling directly impact training dynamics, convergence, and the model's ability to generalize without overfitting.

This study focuses on improving the stability and performance of a sentiment analysis model on an imbalanced dataset of user suggestions. The core of this work is a series of controlled experiments to investigate the effects of batch size, weight decay, and warmup steps. By addressing the class imbalance with a weighted loss function and systematically tuning these hyperparameters, we aim to identify a configuration that maximizes performance and reduces overfitting.

II. Methodology

A. Dataset and Preprocessing

The. The dataset consists of user-provided text suggestions and their corresponding ground truth sentiment labels, categorized into three classes: 0 (Negative), 1 (Neutral), and 2 (Positive).

1. Lowercasing: All text was converted to lowercase to ensure uniformity.
2. Stop Word Removal: A combined set of English stop words from the NLTK library [3] and a custom list of Tagalog stop words were removed.
3. Normalization: URLs, email addresses, phone numbers, and extraneous whitespace were removed or replaced with generic tokens
4. Boilerplate Removal: Non-informative entries such as "none," "n/a," or "wala" were identified and discarded.

Following preprocessing, the dataset was partitioned into a training set (72%), a validation set (18%), and a test set (10%) to be used for model training, hyperparameter tuning, and final evaluation, respectively.

B. Addressing Class Imbalance

An initial analysis of the dataset revealed a significant class imbalance, with a disproportionately large number of neutral samples compared to positive and negative ones. To mitigate the risk of the model becoming biased towards the majority class, a class weighting strategy was employed.

Class weights were calculated using Scikit-learn's compute_class_weight function with the class_weight='balanced' parameter [4]. This function automatically assigns higher weights to minority classes and lower weights to the majority class, inversely proportional to their frequencies. The computed weights were:

- Class 0: 11.40
- *Class 1: 0.43*
- *Class 2: 1.75*

These weights were incorporated into the training process by creating a WeightedLossTrainer, a custom class that inherits from the Hugging Face Trainer [5]. This custom trainer utilizes a torch.nn.CrossEntropyLoss

function initialized with the calculated class weights, ensuring that the model incurs a larger penalty for misclassifying samples from the minority classes.

C. Model and Tokenization

The distilbert-base-multilingual-cased model, a smaller and faster version of BERT, was chosen as the base architecture. It was initialized for sequence classification with three output labels corresponding to the sentiment classes. The accompanying tokenizer was used to convert the preprocessed text into a format suitable for the model. All suggestions were padded or truncated to a maximum length of 128 tokens.

III. Experiments and Results

Four experiments were conducted to evaluate the impact of different hyperparameter configurations on model performance. The evaluation metrics used were Accuracy, and weighted averages for Precision, Recall, and F1-Score.

A. Experiment 1: Initial Experiment

This experiment established a baseline for comparison with moderate settings..

Hyperparameters:

- per_device_train_batch_size: 16
- weight_decay: 0.01
- warmup_steps: 0.

Results: The model achieved a high level of performance on the test set, with a final accuracy and F1-score of 0.990. The classification report showed near-perfect precision for classes 0 and 2. However, the recall for the minority class (0) was 0.90, indicating that the model still misclassified one of the ten negative instances.

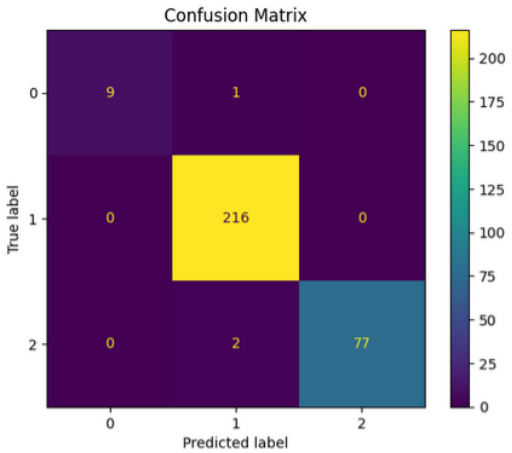

Fig 1. Experiment 1 Confusion Matrix

B. Experiment 2: Smaller Batch Size and No Regularization

This experiment tested the effect of a smaller batch size, which can sometimes offer better generalization, and removed L2 regularization.

Hyperparameters:

- per_device_train_batch_size: 8
- weight_decay: 0.0
- warmup_steps: 0

Results: This configuration resulted in a slight decrease in performance compared to the baseline, achieving a test accuracy and F1-score of 0.987. The validation loss curve showed more fluctuations, and the final validation F1-score at the best epoch (0.9927) was marginally lower than in other experiments. The classification report was identical to Experiment 1, with a recall of 0.90 for the negative class.

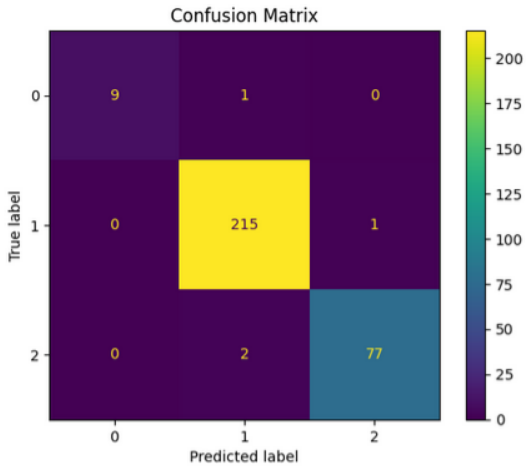

Fig 2. Experiment 2 Confusion Matrix

C. Experiment 3: Introducing Warmup Steps

This experiment investigated the impact of a learning rate warmup schedule, which can help stabilize training in the early epochs.

Hyperparameters:

- per_device_train_batch_size: 16
- weight_decay: 0.01
- warmup_steps: 200

Results: The performance on the test set was identical to Experiment 2, with an accuracy and F1-score of 0.987. The introduction of 200 warmup steps did not

lead to an improvement in final test metrics over the baseline. The validation loss remained higher than in other experiments, suggesting that this particular warmup schedule did not confer a significant stability advantage for this dataset.

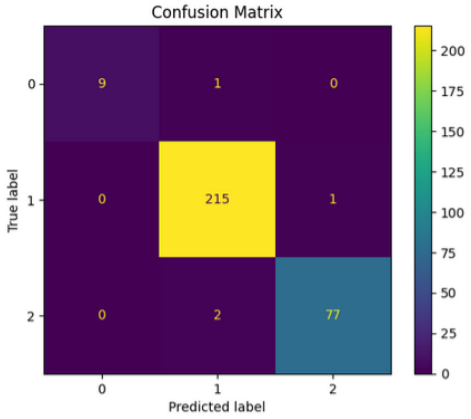

Fig 3. Experiment 3 Confusion Matrix

D. Experiment 4: Larger Batch Size and Stronger Regularization

This experiment explored a larger batch size combined with stronger regularization (higher weight decay) and a longer warmup period.

Hyperparameters:

- per_device_train_batch_size: 32
- weight_decay: 0.1
- warmup_steps: 500

Results: This configuration yielded the best performance during validation. The training logs showed that the model achieved a validation F1-score of 0.9945 at its best epoch (epoch 8), outperforming all other configurations. The validation loss at this epoch was also the lowest recorded across all experiments (0.0453). This suggests that the combination of a larger batch size with stronger regularization and a longer warmup period promoted better learning stability and generalization, effectively reducing overfitting.

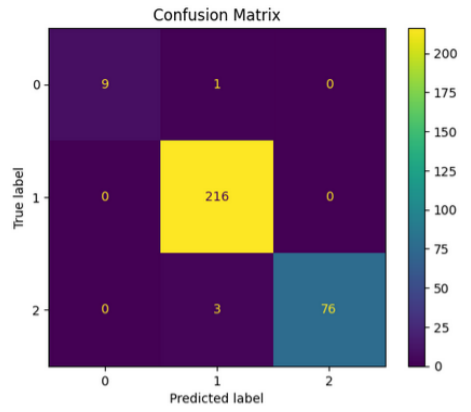

Fig 4. Experiment 4 Confusion Matrix

IV. Discussion And Comparison

A comparative analysis of the four experiments provides insights into the influence of each hyperparameter on the model's training dynamics and final performance. The results are summarized in Table I.

Table 1. Hyperparameter result Comparison

- Impact of Batch Size: Comparing Experiment 2 (batch size 8) with Experiment 1 (16) and Experiment 4 (32) suggests that a larger batch size was beneficial. While smaller batch sizes can sometimes offer a regularizing effect, the larger batch size of 32 in Experiment 4, paired with other optimal parameters, led to the most stable training and the highest validation F1-score.
- Impact of Weight Decay: The role of regularization is critical. Experiment 2, with no weight decay (0.0), performed slightly worse than the baseline. In contrast, Experiment 4, which used a significantly higher weight decay of 0.1, achieved the best validation score. This indicates that stronger L2 regularization was effective in preventing the model from overfitting to the training data, thereby improving its ability to generalize.
- Impact of Warmup Steps: Comparing Experiment 1 (0 warmup steps) with Experiment 3 (200) shows a slight drop in performance on the test set. However, Experiment 4, with 500 warmup steps, achieved the best validation metrics. This suggests that a brief warmup may not be beneficial, but a more substantial warmup period can help stabilize the initial phase of training, especially when using larger batch sizes and stronger regularization, allowing the model to converge to a better solution.

Overall, Experiment 4 demonstrated the most promising configuration. The combination of a large batch size, strong weight decay, and a long warmup period created a stable training environment that led to the best-performing model on the validation set.

V. Conclusion

This paper successfully demonstrated a methodology for fine-tuning a distilbert-base-multilingual-cased model for sentiment analysis on a highly imbalanced dataset. By implementing a weighted loss function and systematically experimenting with batch size, weight decay, and warmup steps, we were able to improve learning stability and achieve excellent performance. The results indicate that a configuration with a batch size of 32, a weight decay of 0.1, and 500 warmup steps provides the best generalization capabilities for this specific task. This study underscores the importance of both addressing class

imbalance at the loss function level and conducting rigorous hyperparameter tuning to unlock the full potential of pre-trained transformer models. Future work could explore other models, advanced data augmentation techniques for the minority classes, or a more exhaustive hyperparameter search using automated tools.

VI. Conclusion

[1] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008.

[2] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," in Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing, 2019.

[3] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, Inc., 2009.

[4] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[5] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38-45.