Fine-tuning Transformers for Emotion Detection: A Comparative Analysis

Nguyen Cao Duy
A0258078R
nguyen.cao.duy@u.nus.edu

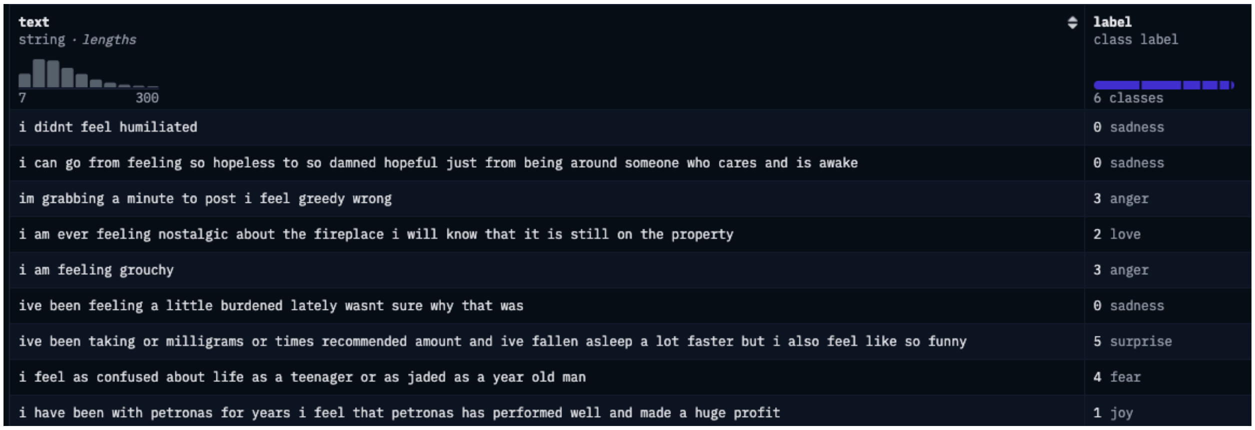
Abstract This report investigates strategies for adapting large pretrained transformer models to the downstream task of text-based emotion detection. We compare traditional full fine-tuning of a RoBERTa-base model with a more parameter-efficient method, Prompt Tuning. Through a series of controlled experiments on the CARER emotion dataset, we analyze the performance differences and efficiency trade-offs between these two approaches. Our findings highlight the critical role of the learning rate in both methods and demonstrate that prompt tuning, while requiring significantly fewer trainable parameters, can achieve competitive performance. We also explore the impact of prompt initialization, showing that prompts initialized with task-relevant text converge faster than randomly initialized prompts, offering valuable insights into efficient model adaptation techniques.

1. Introduction

Large Language Models (LLMs) like RoBERTa (Liu et al. 2019), pretrained on vast corpora of text, have demonstrated remarkable capabilities across a wide range of natural language understanding tasks. However, adapting these massive models to specific, domain-focused problems requires careful fine-tuning. This report explores and compares two primary fine-tuning strategies: traditional full fine-tuning, where all model parameters are updated, and Prompt Tuning, a Parameter-Efficient Fine-Tuning (PEFT) method that freezes the model backbone and only trains a small number of prefixed "virtual" or "soft" tokens (Lester, Al-Rfou, and Constant 2021).

The goal of this work is to analyze the performance differences, efficiency trade-offs, and practical lessons learned when applying these methods to the task of emotion detection.

2. Dataset: Emotion Detection



text	label
string · lengths 7 300	class label 6 classes
i didnt feel humiliated	0 sadness
i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	0 sadness
im grabbing a minute to post i feel greedy wrong	3 anger
i am ever feeling nostalgic about the fireplace i will know that it is still on the property	2 love
i am feeling grouchy	3 anger
ive been feeling a little burdened lately wasnt sure why that was	0 sadness
ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny	5 surprise
i feel as confused about life as a teenager or as jaded as a year old man	4 fear
i have been with petronas for years i feel that petronas has performed well and made a huge profit	1 joy

Figure 1: Examples from the CARER emotion dataset, from `dair-ai/emotion` on Hugging Face.

For this study, we selected the CARER emotion dataset (Saravia et al. 2018), a widely used benchmark for multi-class single-label text classification. It consists of English text samples primarily sourced from Twitter. We utilized the predefined train (16k), validation (2k), and test (2k) splits.

Motivation & Domain: The task is to classify a given text into one of six distinct emotional categories: sadness, joy, love, anger, fear, and surprise. This is a relevant and challenging task in domains such as social media analysis, customer feedback processing, and mental health monitoring. An example of the data is shown in Figure 1.

3. Experimental Setup

To ensure a fair comparison between fine-tuning methods, we set up a consistent experimental framework.

Fixed Parameters: All experiments were conducted using the `roberta-base` model on Hugging Face, with around **125 million parameters**. Training was conducted on a single **NVIDIA H100 GPU** with 96GB of memory through the SoC Compute Cluster. We trained for a fixed **5 epochs** with a **batch size of 64**. We used the **AdamW optimizer** with default beta and weight decay parameters, as well as a **linear learning rate scheduler** with a default warmup period. Evaluation was performed on the validation set at the end of each epoch, and the best checkpoint was saved and loaded for prediction on the test set at the end. The primary evaluation metric was **accuracy**. To ensure reproducibility, a global seed of `23` was used for all runs.

Hyperparameter Tuning: For each method, we performed a systematic hyperparameter sweep using Weights & Biases to identify the optimal configuration. The best model was selected based on the highest `eval/accuracy` achieved on the validation set during training.

4. Results and Analysis

This section details the results of our experiments, focusing on performance differences, efficiency trade-offs, and key lessons learned as required by the assignment.

4.1. Full Fine-Tuning: The Baseline

Our first experiment established a baseline by fully fine-tuning all 125 million parameters of the `roberta-base` model. We swept across four learning rates: `2e-5`, `2e-4`, `2e-3`, and `2e-2`.

A critical finding was that the learning rate needs to be very small when fine-tuning the entire model. As shown in Figure Figure 2, only the lowest learning rate of `2e-5` led to successful convergence, achieving a validation accuracy exceeding 90%. The higher learning rates (`2e-4`, `2e-3`, `2e-2`) resulted in unstable training; the model's loss did not decrease, and its accuracy remained at random guessing levels, indicating catastrophic divergence. This highlights the importance of using a small, carefully chosen learning rate when updating all parameters of a large transformer model.

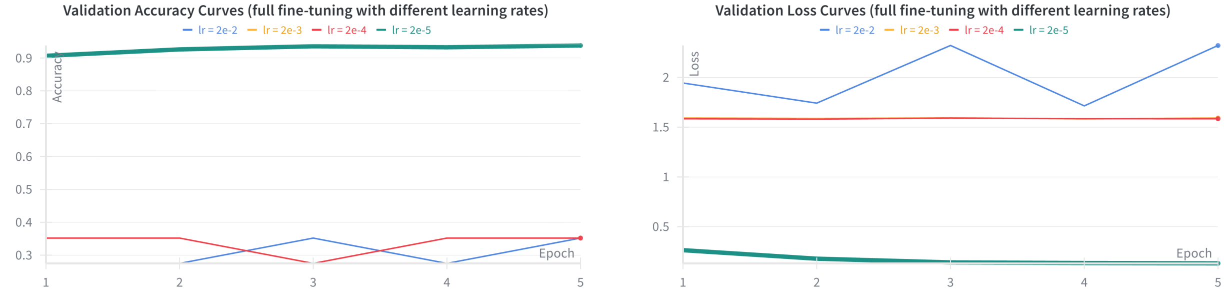


Figure 2: Validation curves for full fine-tuning across different learning rates. Full fine-tuning requires very low learning rates at the scale of 2e-5 to converge effectively.

4.2. Prompt Tuning: An Efficient Alternative

Next, we investigated Prompt Tuning, a PEFT method that freezes the base model and only trains a small set of virtual tokens. We explored two initialization variants:

1. **Random Initialization:** Virtual tokens are initialized from a random distribution. We swept across 4 learning rates and 3 different numbers of virtual tokens (`8`, `16`, `32`), for a total of 12 runs.
2. **Text Initialization:** Virtual tokens are initialized from the embeddings of task-relevant text. This provides a stronger starting point. We swept across the same 4 learning rates and 3 different initialization texts of similar length (12 tokens) but different levels of relevance to the task, for a total of 12 runs. The texts used were:
   - “classify the emotion of the following text: “ (highly relevant)
   - “assign the following text into suitable categories: “ (moderately relevant)
   - “read the following text and think about it: “ (less relevant)

From these experiments, we derived two key observations:

Observation 1: Prompt Tuning Requires Much Higher Learning Rates

Unlike full fine-tuning, prompt tuning benefits from significantly higher learning rates. As shown in Figure 3, lower learning rates like `2e-5` led to severe underfitting within 5 epochs. Optimal performance was achieved with learning rates of `2e-3` or `2e-2`, an order of magnitude higher than the baseline. This might be because only a small subset of parameters (the virtual tokens) are being updated, requiring a more aggressive learning rate to make meaningful progress.

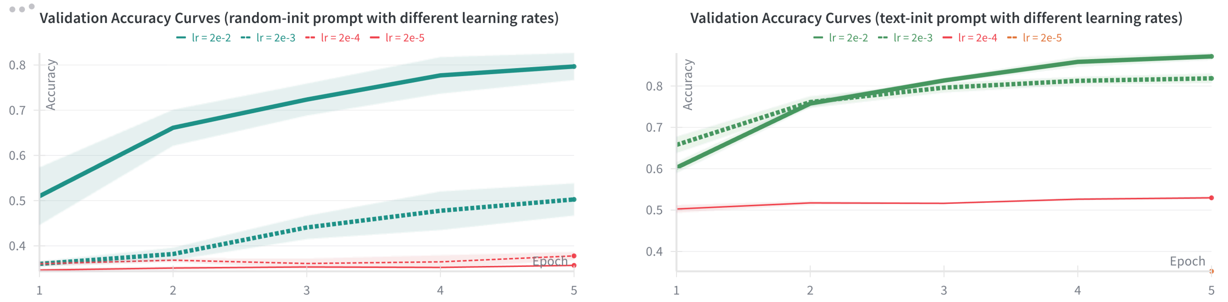


Figure 3: Validation curves for Prompt Tuning across different learning rates. Higher learning rates at the scale of 2e-2 are necessary for fast convergence.

Observation 2: Text Initialization Accelerates Convergence

Our second finding highlights the benefit of initializing virtual tokens with meaningful text. Figure 4 compares the validation accuracy curves of models trained with randomly initialized prompts versus text-initialized prompts at the same learning rate of `2e-2`.

For random initialization, as the number of virtual tokens increased from `8` to `32`, the model’s performance improved, indicating that more capacity helps.

However, this improvement came at the cost of more training parameters and was still outperformed by text-initialized prompts with only `12` tokens. The text-initialized models, regardless of the specific prompt text used, consistently achieved higher accuracy faster than the best random initialization setup with `32` tokens.

The relevance of the initialization text also played a role. The most relevant prompt (“classify the emotion of the following text: “) led to the fastest convergence and highest accuracy, while less relevant prompts resulted in slightly slower learning curves. This suggests that providing task-relevant context in the prompt initialization can significantly enhance learning efficiency.

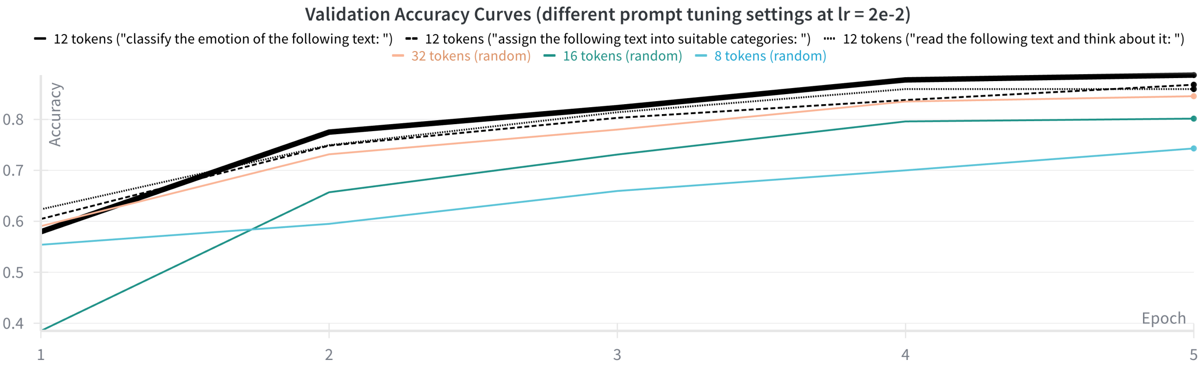


Figure 4: Validation accuracy of random vs. text initialization for prompts. Relevant text initialization leads to faster convergence.

4.3. Performance and Efficiency Trade-offs

The core of our analysis is the direct comparison between the best-performing full fine-tuning model and the best prompt tuning model. Table 1 encapsulates the key trade-offs.

While full fine-tuning achieved the highest absolute accuracy on the test set, prompt tuning delivered a competitive result (within 9% accuracy drop) while training over **200x fewer parameters**. This drastic reduction in trainable parameters also translated to a noticeable decrease in training time (about 20% time reduction for text initialization and 40% for random initialization) as well as a significant reduction in model footprint (over 99% smaller).

Surprisingly, random initialization with 32 tokens consumed less training time than text initialization with 12 tokens, which might be due to the overhead of tokenizing the initialization text.

It also appears that the number of trainable parameters hardly changed with different number of virtual tokens, indicating that we might be able to achieve a better performance closer to the full fine-tuning baseline with more virtual tokens without significant computational cost.

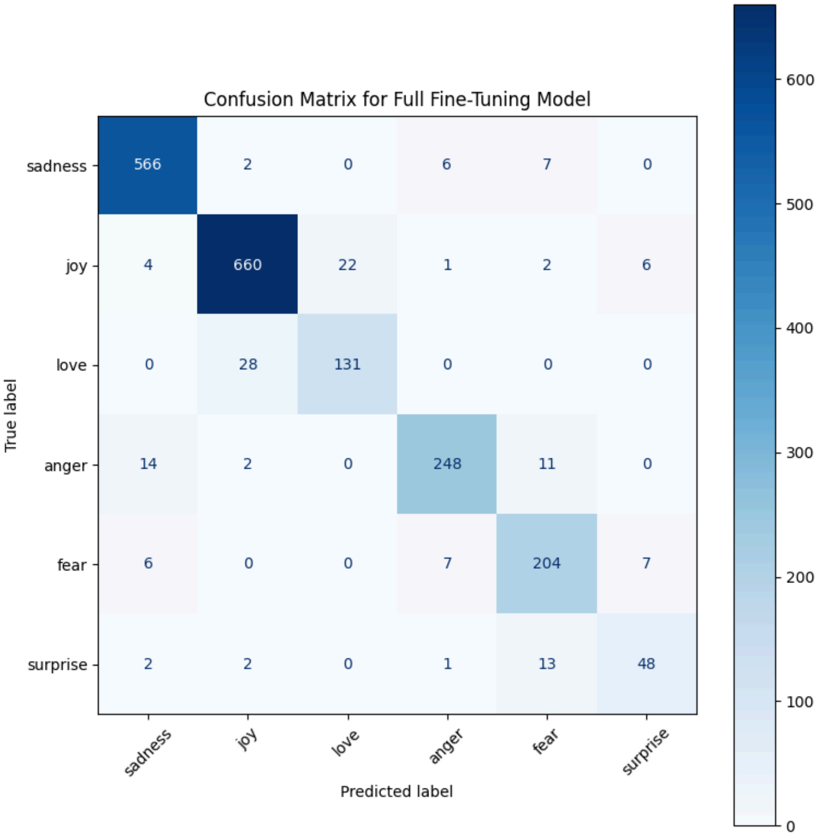
These results clearly demonstrate the efficiency of PEFT methods. For applications where storage, memory, or training time are constrained, prompt tuning offers a highly effective solution with only a minor compromise in performance. The lesson learned is that it is not always necessary to update every weight in a large model to successfully adapt it to a new task.

4.4. Error Analysis

To understand the limitations of our best-performing model (full fine-tuning), we analyzed its errors on the test set. A confusion matrix, shown in Figure 5, reveals the primary sources of misclassification.

Method	**Test Accuracy (%)**	**Trainable Parameters (millions / % of baseline)**	**Model Weight (MB / % of baseline)**	**Training Time (seconds / % of baseline)**
Full Fine-Tuning	92.85	125 / 100%	480, 100%	281 / 100%
Prompt Tuning (32 random tokens)	84.1	0.62 / 0.495%	3.2, 0.67%	**169 / 60.1%**
Prompt Tuning (12 text tokens “classify the emotion of the following text: “)	**87.6**	**0.61 / 0.487%**	**2.3 / 0.48%**	235 / 83.6%

Table 1: Comparison of the best models from each fine-tuning method.



	sadness	joy	love	anger	fear	surprise
sadness	566	2	0	6	7	0
joy	4	660	22	1	2	6
love	0	28	131	0	0	0
anger	14	2	0	248	11	0
fear	6	0	0	7	204	7
surprise	2	2	0	1	13	48

Figure 5: Confusion matrix for the best full fine-tuning model on the test set.

From the matrix, we observe that the most frequent errors occur between semantically similar emotions. For example, the model often confuses **love** with **joy**. This suggests the model struggles with nuanced emotional expressions where lexical cues might overlap.

Text	True Label	Predicted Label	Possible Reason for Error
"i feel a bit stressed even though all the things i have going on are fun"	`anger`	`sadness`	Mixed feelings. Mentions stress (anger cue) but positive clause ("fun") may have confused the model toward sadness.
"i feel so blessed and honored that we get to be its parents"	`love`	`joy`	High semantic overlap. "Blessed" is strongly associated with both `love` and `joy`, making it a difficult distinction.
"i feel agitated and annoyed more than worried or fearful but these feelings can easily lead to being short tempered with my family and feelings of disharmon"	`fear`	`anger`	Contrastive phrasing. Explicitly compares `fear` and `anger`, which can confuse classification since both cues co-occur in the same sentence.

Table 2: Qualitative examples of misclassifications.

Table 2 provides a qualitative look at some of these error cases. These examples often involve ambiguous phrasing, or overlapping emotional vocabulary that can mislead the model. For instance, sentences expressing mixed feelings or using words associated with multiple emotions pose significant challenges.

5. Conclusion

In this report, we successfully fine-tuned a RoBERTa-base model for emotion detection using both full fine-tuning and prompt tuning. Our experiments confirm that while full fine-tuning yields the highest performance, it is computationally expensive, especially regarding memory usage. In contrast, prompt tuning provides a powerful and efficient alternative, achieving comparable accuracy by training only a minuscule fraction of the parameters. We demonstrated that prompt tuning requires higher learning rates and that performance can be further accelerated by initializing virtual prompts with task-relevant text. These findings highlight the significant practical advantages of PEFT methods in adapting large models to specialized tasks.

6. Appendix

6.1. Reproducibility

- Github Repository: https://github.com/ncduy0303/dsa4213-assignment-3
- Weights & Biases Project: https://wandb.ai/ncduy0303/dsa4213-assignment-3
- Google Drive Folder (for model weights): https://drive.google.com/drive/folders/1fu7JaSRZgzYIWQrdMngBoHxRez4NEHxz?usp=sharing

6.2. AI Tool Declaration

I used Gemini 2.5 Pro and Github Copilot to generate ideas, format paragraphs, improve drafts, refine, and finalize the assignment. I am responsible for the content and quality of the submitted work.

7. References

Lester, Brian, Rami Al-Rfou, and Noah Constant. 2021. “The Power of Scale for Parameter-Efficient Prompt Tuning”. 2021. https://arxiv.org/abs/2104.08691.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. “Roberta: A Robustly Optimized BERT Pretraining Approach”. 2019. https://arxiv.org/abs/1907.11692.

Saravia, Elvis, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. “CARER: Contextualized Affect Representations for Emotion Recognition”. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3687–97. Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1404.