

Homework 2 (CSCI 5541 NLP)

Anurag Purohit
puroh029@umn.edu

Abstract

I have fine-tuned pre-trained transformers for binary sentiment on GLUE/SST-2 using a custom training loop (AdamW, cross-entropy, dynamic padding, gradient clipping) on a Colab L4 GPU. I compared DistilBERT-base-uncased and BERT-base-uncased. I achieved best validation accuracies of 0.90 and 0.916, with mild overfitting after epochs 3-4. I analyzed misclassified validation samples with confidences and found recurring issues from negation/contrast, sarcastic or mixed tone, and spurious lexical shortcuts (e.g., names/genres). I also prompted a small instruction-tuned LLM on these errors—sometimes correcting negation but missing domain cues. I have included code, plots, and an error CSV for reproducibility.

1 Introduction

Text classification is a core NLP task with broad applications (search, moderation, analytics). In this project I tackle binary sentiment classification on the GLUE/SST-2 benchmark using pre-trained transformer encoders. Rather than relying on Hugging Face’s default training loop, I have implemented a custom fine-tuning pipeline that handles tokenization, dynamic padding, loss/optimizer setup, gradient-norm clipping, and per-epoch evaluation. My focus is to (1) quantify how much accuracy a compact model can achieve with careful training on modest hardware, and (2) understand why the model still fails on certain inputs.

I have fine-tuned and compared DistilBERT-base-uncased and BERT-base-uncased on a Colab L4 GPU, using accuracy as the primary metric. I have tracked learning curves to verify correct optimization and detect overfitting, then perform an error analysis by collecting misclassified validation samples with confidence scores. To contextualize results, I have also prompted a small instruction-tuned LLM on the same errors to see when a zero-shot model helps or hurts relative to a fine-tuned

encoder.

Contributions and deliverables.

1. A clean, reproducible fine-tuning script with a custom training loop (AdamW, CE loss, dynamic padding, gradient clipping).
2. A head-to-head evaluation of DistilBERT vs. BERT on SST-2 with learning curves and final accuracy.
3. A spreadsheet of ≥ 20 misclassified samples with model confidences and a qualitative analysis of common failure modes (negation/contrast, sarcasm, spurious lexical cues).
4. A brief comparison to an instruction-tuned LLM on the same hard cases.

The rest of the report describes the dataset and setup, training procedure, results and comparisons, error analysis, and takeaways.

2 Related Work

Pre-trained transformer encoders have become the default for transfer learning in NLP, where a general language model is fine-tuned on a downstream task such as sentiment classification. I have built on this line of work by comparing BERT-base-uncased and the lighter DistilBERT-base-uncased on SST-2. I have treated the official GLUE validation set as my evaluation split (as the official test set lacks labels).

3 Task and Dataset

Task - Binary sentiment classification: given a single sentence, predict positive or negative.

Dataset - GLUE/SST-2. I have used the Hugging Face datasets version: 67k training sentences and 872 validation sentences, with labels 0,1. I have tokenized with the model’s WordPiece tokenizer and truncating sequences to the model’s max length,

and relying on dynamic padding at batch time (via `DataCollatorWithPadding`). This was done to avoid padding every example to a global max. The primary metric is accuracy on the validation split.

4 Setup/Configuration

- Software- Python 3, PyTorch + Hugging Face Transformers/Datasets/Evaluate.
- Hardware- Google Colab with a single L4 GPU.
- Reproducibility- Seeded Python/NumPy/PyTorch/CUDA.

5 Methods

Models:

- DistilBERT-base-uncased ($\approx 66\text{M}$ parameters).
- BERT-base-uncased ($\approx 110\text{M}$ parameters).

Each gets a randomly initialized classification head (linear layer over the [CLS] representation).

Training loop- I implemented a custom loop (by subclassing `Trainer` and overriding the inner loop) that performs: tokenization \rightarrow dynamic padding \rightarrow forward \rightarrow cross-entropy loss \rightarrow backward \rightarrow gradient-norm clipping (`clip_grad_norm_=1.0`) \rightarrow AdamW update \rightarrow per-epoch evaluation. I also use a simple linear LR schedule and set a fixed random seed for reproducibility.

Hyperparameters-

- batch size 32 (train) / 64 (eval),
- learning rate $5\text{e-}5$,
- epochs 5,
- weight decay $1\text{e-}4$,
- max sequence length = model default.

I ran everything on a Colab NVIDIA L4 GPU.

6 Results

Learning behavior. Training loss decreases monotonically; validation loss bottoms out around epochs 3-4, then creeps up—classical mild overfitting.

Accuracy:

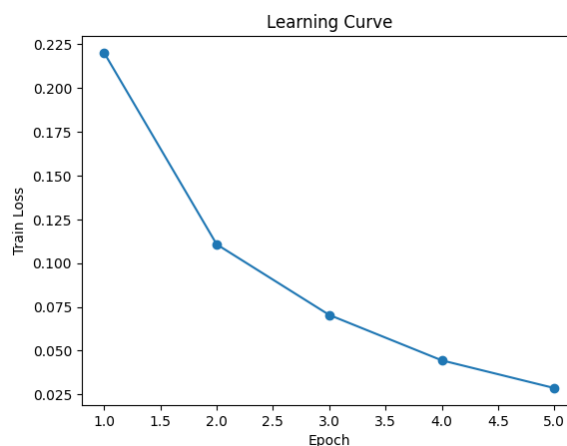


Figure 1: Training loss learning curve

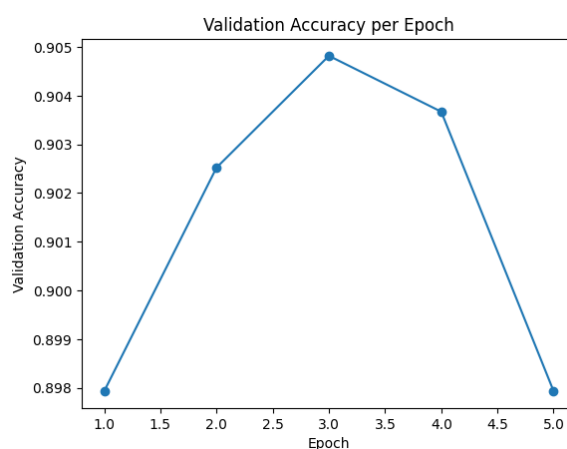


Figure 2: Validation accuracy plot

- DistilBERT-base-uncased: best val acc ≈ 0.905 (peak around epoch 3; final printed value at epoch 5 was 0.898).
- BERT-base-uncased: best val acc ≈ 0.919 (peak at epoch 1; final printed value at epoch 5 was 0.916).
- BERT is consistently stronger but $2\times$ slower on L4.

Timing- DistilBERT 11.8 min total for 5 epochs; BERT-base 22.2 min.

Takeaway- With minimal tuning, BERT-base reaches 0.92 val accuracy on SST-2; DistilBERT trails slightly but remains competitive at 0.90 with roughly half the wall-time. Selecting the best validation epoch (early stopping) would modestly improve the reported number for both models versus the last epoch.

6.1 Error Analysis

I exported a .csv file of misclassified validation samples with the model’s softmax confidence. Reviewing ≥ 20 items, I observed recurring failure modes:

1. Negation/contrast the model under-represents (e.g., “I wanted to like it, but. . .”).
2. Sarcastic/mixed tone, where positive cue words appear inside an overall negative stance.
3. Language shortcuts (proper nouns/genre words) that correlate with one label in training and mislead the classifier on edge cases.
4. High-confidence mistakes: many errors have $\hat{p}(y) > 0.9$, suggesting over-confidence typical of cross-entropy without calibration.

Possible improvements- Early stopping at the best val epoch; modest regularization (weight decay already on, adding small dropout on the head could help); simple text cleaning (e.g., HTML breaks); Longer training or trying a RoBERTa-base backbone.

6.2 LLM Comparison

I also prompted a small instruction-tuned LLM on a subset of 20 errors made by the fine-tuned model. Qualitatively, the LLM sometimes fixes negation/contrast cases but struggles with domain-specific references and nuanced film-critic phrasing. This suggests complementary strengths: the encoder excels at in-distribution lexical patterns; the LLM occasionally helps with pragmatic inferences but is not uniformly superior on short, single-sentence SST-2 items. (I have included a CSV with the 20 texts, ground truth label, fine-tuned prediction+confidence, and the LLM’s prediction.)

7 Limitations

SST-2’s validation set is small; a single run can vary by ~ 0.5 – 1.0 accuracy points. I intentionally kept training to 5 epochs for compute fairness; stronger results are achievable with tuned LR schedules, longer training with early stopping, or larger backbones. I did not calibrate probabilities, so confidence values should be interpreted cautiously.

8 Conclusion

I fine-tuned DistilBERT-base and BERT-base on SST-2 with a custom training loop featuring dynamic padding and gradient clipping. On L4, BERT-base achieved 0.919 best validation accuracy (DistilBERT 0.905) with mild overfitting after 3–4 epochs. Error analysis highlights negation/contrast, sarcasm, and spurious lexical cues as dominant failure modes. A small instruction-tuned LLM sometimes repairs negation errors but is inconsistent on domain-specific language. Early stopping and light regularization are practical improvements; with more time, larger backbones or better calibration would further refine results.

References

- Quentin Lhoest, Clement Delangue, Patrick von Platen, et al. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*.