Comparing Pretrained Transformer Models on the GLUE SST-2 Sentiment Classification Task

Ye Cai
Department of Statistics
University of Michigan
Email: cyea@umich.edu

***Abstract*—Pretrained Transformer language models have become the standard backbone for many natural language processing tasks. In this project we fine-tune and compare four widely used pretrained encoders—DistilBERT, BERT-base-uncased, ALBERT-base-v2, and RoBERTa-base—on the SST-2 sentiment classification task from the GLUE benchmark. Using the Hugging Face transformers and datasets libraries in a unified Python pipeline, we keep preprocessing, optimization hyperparameters, and the evaluation protocol fixed so that performance differences can be attributed mainly to the model architectures. We report validation accuracy, training runtime, and throughput on a single T4 GPU, and analyze the training and evaluation curves logged with Weights and Biases. RoBERTa-base attains the best validation accuracy (around 93.3%), while DistilBERT trains substantially faster at the cost of several points of accuracy. BERT-base and ALBERT-base-v2 lie between these extremes. Our results quantify the accuracy–efficiency trade-off among these popular models and provide practical guidance for choosing a pretrained encoder for resource-constrained sentiment classification applications.**

***Index Terms*—Transformers, sentiment analysis, text classification, GLUE, SST-2, Hugging Face**

I. Introduction

Sentiment classification aims to automatically determine whether a piece of text expresses a positive or negative opinion. It is a core component in applications such as product review mining, social media monitoring, and user feedback analysis. Large pretrained Transformer language models have recently become the standard approach for such tasks by providing contextual representations that can be adapted to new datasets with relatively little labeled data.

Among the most influential models are BERT, RoBERTa, ALBERT, and DistilBERT. BERT-base [1] introduced bidirectional Transformer encoders pretrained with masked language modeling and next-sentence prediction, and quickly became a strong baseline on the GLUE benchmark. RoBERTa [2] showed that with more data and a better training recipe, the same architecture can reach even stronger performance. ALBERT [3] reduces memory footprint and improves parameter efficiency via factorized embeddings and cross-layer parameter sharing. DistilBERT [4] applies knowledge distillation to obtain a smaller and faster model while retaining most of BERT's accuracy. All of these models are available through the Hugging Face Transformers library and are widely used as backbones for text classification.

While benchmark leaderboards often focus on the best achievable accuracy, practitioners also care about the trade-off between performance and efficiency under realistic resource constraints. A heavier model may yield slightly better accuracy but require longer training time and higher inference cost, whereas a lighter model may be easier to deploy but lose several points of accuracy. Understanding these trade-offs on a concrete task is important for making informed model choices in practice.

In this project we focus on the SST-2 sentiment classification task from the GLUE benchmark and ask:

- How do four popular pretrained encoders—DistilBERT, BERT-base-uncased, ALBERT-base-v2, and RoBERTa-base—compare in terms of validation accuracy under a shared fine-tuning pipeline?
- How do their training runtimes on a single T4 GPU differ, and what efficiency gains do the smaller models offer?
- What practical accuracy–efficiency trade-off emerges from these results, and how might it guide model selection for resource-constrained sentiment classification?

To answer these questions, we build a unified fine-tuning pipeline in Python using the Hugging Face Transformers and Datasets libraries. We keep data preprocessing and optimization hyperparameters fixed across models so that differences in performance can be attributed primarily to the architectures. We then analyze both the final validation metrics and the training dynamics logged with Weights and Biases to obtain a detailed comparison of the four models.

II. Method

A. Problem Formulation

We consider binary sentiment classification on single sentences. Let x denote an input sentence and $y \in \{0, 1\}$ denote its sentiment label, where 0 corresponds to negative and 1 to positive sentiment. Given a labeled training set, our goal is to learn a classifier that maps each sentence to the correct label. We model this classifier as a pretrained Transformer encoder followed by a task-specific classification head, and fine-tune all parameters on the SST-2 training set.

B. Dataset

We use the Stanford Sentiment Treebank 2 (SST-2) [6] as provided in the GLUE benchmark [5]. SST-2 contains movie review sentences labeled as expressing either positive or negative sentiment. Following the GLUE setup, we use the official train and validation splits and treat the task as binary single-sentence classification. After filtering out neutral examples, the training set contains approximately 67k sentences and the validation set about 0.9k sentences. We load the data via the Hugging Face datasets library, which returns separate train and validation splits.

C. Models

We compare four pretrained Transformer encoders, all accessed via the Hugging Face Transformers library:

- DistilBERT-base-uncased [4]: a distilled version of BERT-base with fewer layers and parameters, designed to be smaller and faster while retaining most of BERT's performance.
- BERT-base-uncased [1]: the original 12-layer Transformer encoder pretrained with masked language modeling and next-sentence prediction.
- ALBERT-base-v2 [3]: a parameter-efficient variant of BERT that factorizes the embedding matrix and shares weights across layers.
- RoBERTa-base [2]: an optimized BERT-style model trained on more data with a modified pretraining objective and larger batches.

For each model we use the corresponding tokenizer and sequence-classification head. The classification layer is randomly initialized and maps the pooled representation to two output logits.

D. Training Setup

1) Preprocessing: We tokenize each sentence using the model-specific tokenizer, truncate sequences longer than 128 tokens, and apply padding so that all examples in a mini-batch have the same length. The preprocessing function is applied to the whole dataset with the datasets map method, and the tokenized results are cached to avoid repeated computation.

2) Optimization: To enable a fair comparison, we adopt a shared set of fine-tuning hyperparameters for all four models. Unless otherwise noted, we train for 5 epochs with a batch size of 16, using the AdamW optimizer with a learning rate of 2×10^{-5} and weight decay of 0.01. We use a linear learning rate schedule with warmup, evaluate on the validation set at the end of each epoch, and select the checkpoint with the best validation accuracy as the final model.

We implement training using the Trainer API from Hugging Face Transformers, which handles the training loop and evaluation. Accuracy is used as the primary metric for SST-2. All experiments are run on a single NVIDIA T4 GPU in Google Colab. During training, we log loss, accuracy, and other diagnostics to Weights and Biases and later use these logs to analyze training dynamics and compare efficiency across models.

...

	model	eval_accuracy	train_runtime
0	albert-base-v2	0.924312	2501.4015
1	bert-base-uncased	0.923165	2702.7464
2	distilbert-base-uncased	0.909406	1741.1033
3	roberta-base	0.933486	2964.0291

Fig. 1. Validation accuracy of four pretrained models on the SST-2 dev set.

TABLE I
VALIDATION ACCURACY AND TRAINING RUNTIME ON SST-2.

Model	Dev accuracy	Train runtime (s)
ALBERT-base-v2	0.9243	2501.4
BERT-base-uncased	0.9232	2702.7
DistilBERT-base	0.9094	1741.1
RoBERTa-base	0.9335	2964.0

III. RESULTS

A. Overall Validation Performance

Table I and Figure 1 summarize the validation accuracy of the four pretrained models on SST-2. RoBERTa-base achieves the best performance with an accuracy of about 93.35%. ALBERT-base-v2 and BERT-base-uncased are close behind at around 92.3–92.4%, while DistilBERT is lower at roughly 90.94%, reflecting the cost of model compression in this setting.

B. Training Efficiency

Figure 2 compares the total training time of the models on a single T4 GPU. DistilBERT is the fastest model, finishing training in roughly 1 700 seconds, while RoBERTa-base is the slowest at about 3 000 seconds. BERT-base and ALBERT-base-v2 lie in between. Together with the accuracy results, this shows a clear accuracy–efficiency trade-off: RoBERTa-base provides the best accuracy at the highest cost, whereas DistilBERT offers the shortest training time but loses a few points of accuracy.

C. Training Dynamics

We further inspect the training and evaluation curves logged to Weights and Biases. Figure 3 shows the training loss as a function of global steps. All four models exhibit a smooth decrease and appear to converge within the 5 training epochs, with RoBERTa-base and BERT-base-uncased reaching slightly lower final loss values than ALBERT-base-v2 and DistilBERT.

Figure 4 presents the evaluation accuracy measured at several checkpoints during training. RoBERTa-base consistently matches or outperforms the other models, while DistilBERT remains below the heavier encoders throughout training.

The evaluation loss curves in Figure 5 decrease rapidly during the first few thousand steps and then plateau, indicating that five epochs of fine-tuning are sufficient for this task and that none of the models shows strong signs of overfitting.

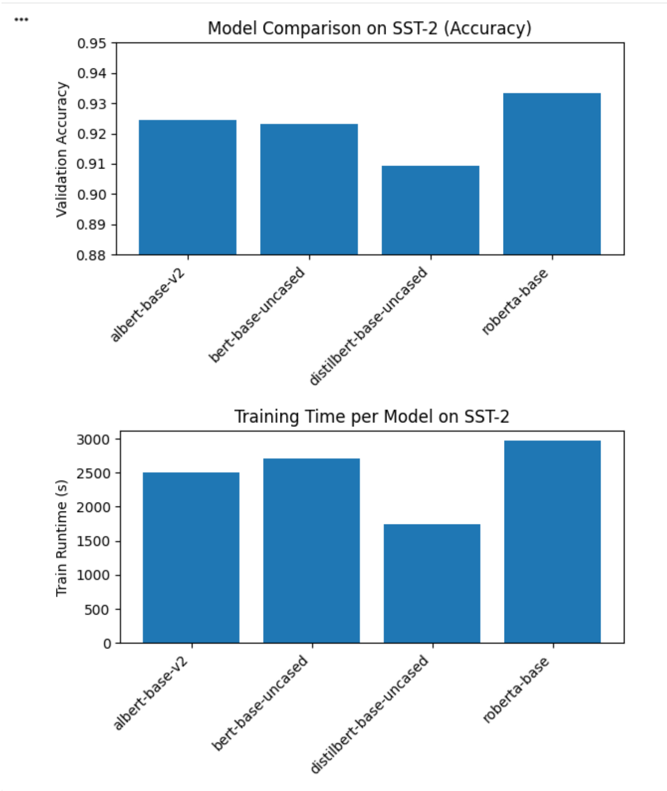

Fig. 2. Training time (in seconds) of each model on SST-2.

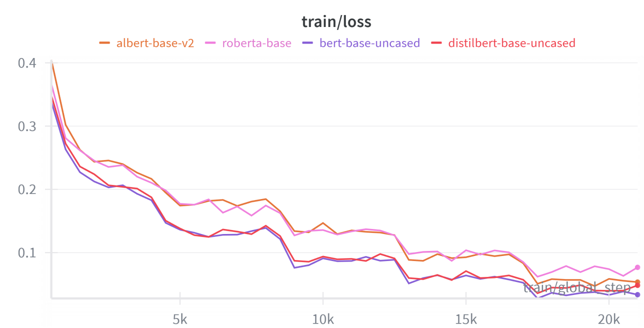

Fig. 3. Training loss versus global steps for the four models (exported from Weights and Biases).

D. Label Distribution

Finally, Figure 6 visualizes the label distribution in the SST-2 training set. The dataset is slightly imbalanced with more positive than negative examples, but the imbalance is not extreme and all models are able to achieve high accuracy despite this mild skew.

IV. Conclusion

In this project we built a unified fine-tuning pipeline with the Hugging Face Transformers and Datasets libraries and used it to compare four pretrained encoders—DistilBERT, BERT-base-uncased, ALBERT-base-v2, and RoBERTa-base—on the SST-2 sentiment classification task from the GLUE

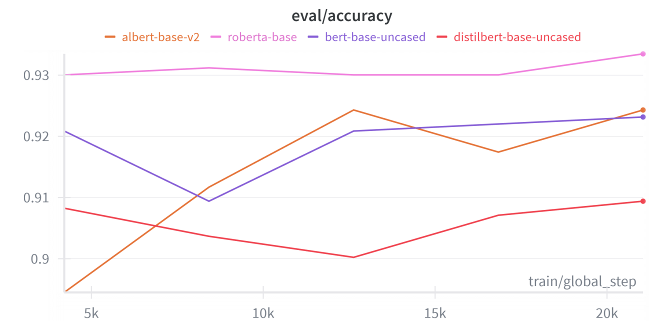

Fig. 4. Evaluation accuracy over training for the four models.

Fig. 5. Evaluation loss over training for the four models.

benchmark. By keeping preprocessing, optimization hyperparameters, and the evaluation protocol fixed across models, we attribute differences in downstream performance primarily to the underlying architectures and pretraining strategies.

Our experiments show that RoBERTa-base achieves the best validation accuracy on SST-2 (about 93.3%), confirming its strong performance reported in the literature. ALBERT-base-v2 and BERT-base-uncased obtain very similar accuracies around 92.3–92.4%, suggesting that parameter sharing in ALBERT can reduce memory usage without noticeably harming performance on this task. DistilBERT, the smallest model considered, is several points worse at roughly 90.9% accuracy, but it trains substantially faster than the larger encoders. The training runtime measurements and W&B curves make the resulting accuracy–efficiency trade-off explicit: heavier models deliver modest but consistent gains in accuracy at the cost of longer training time and higher computational requirements.

These findings suggest the following practical guidance for sentiment classification on SST-2-like data. When accuracy is the primary objective and resources permit, RoBERTa-base is a strong default choice. When memory or time is constrained, ALBERT-base-v2 or BERT-base-uncased provide a good compromise between performance and efficiency. DistilBERT is attractive in scenarios where fast experimentation or deployment on limited hardware matters more than achieving the very best accuracy.

This study has several limitations. We only considered a

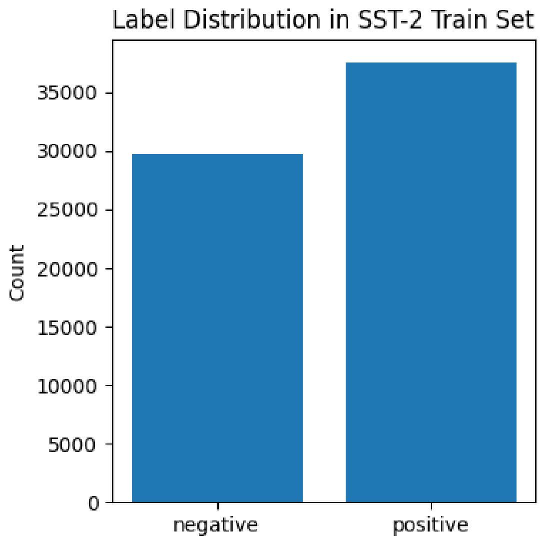

Fig. 6. Label distribution of the SST-2 training set.

single GLUE task and a single set of fine-tuning hyperparameters, and we measured efficiency in terms of training runtime on one T4 GPU rather than full inference latency or energy usage. Future work could extend this comparison to additional GLUE tasks, larger models (e.g., BERT-large and RoBERTa-large), and more aggressive hyperparameter tuning or learning-rate schedules, as well as evaluating distilled and quantized variants under strict latency or memory budgets. Nevertheless, the present results provide a concrete, reproducible case study of how different pretrained Transformer encoders behave on a standard sentiment classification benchmark.

V. References

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," *arXiv preprint* arXiv:1909.11942, 2019.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT: A distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[5] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, 2018, pp. 353–355.

[6] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013.