Does Fine-tuning a Classifier Help in Low-budget Scenarios? Not Much

Cesar Gonzalez-Gutierrez, Audi Primadhanty, Francesco Cazzaro, Ariadna Quattoni
Universitat Politècnica de Catalunya, Barcelona, Spain
{cesar.gonzalez.gutierrez, audi.primadhanty, francesco.cazzaro}@upc.edu,
aquattoni@cs.upc.edu

Abstract

In recent years, the two-step approach for text classification based on pre-training plus fine-tuning has led to significant improvements in classification performance. In this paper, we study the low-budget scenario, and we ask whether it is justified to allocate the additional resources needed for fine-tuning complex models. To do so, we isolate the gains obtained from pre-training from those obtained from fine-tuning. We find out that, when the gains from pre-training are factored out, the performance attained by using complex transformer models leads to marginal improvements over simpler models. Therefore, in this scenario, utilizing simpler classifiers on top of pre-trained representations proves to be a viable alternative.

1 Introduction

In the past few years, a dominating paradigm has emerged in text classification, primarily centered on a two-step approach: inducing pre-trained weights, followed by task fine-tuning using a transformer model with supervised labeled data (Radford et al., 2018; Devlin et al., 2019). The new approach has led to significant improvements over previous classification strategies based on simpler linear models trained on sparse bag-of-words feature representations.

The improvements observed in performance are often attributed to the induced representation (Miaschi and Dell'Orletta, 2020; Talmor et al., 2020; Xia et al., 2020). It is not surprising that leveraging contextual continuous word embeddings can lead to improvements by mitigating the sparsity issues of classical bag-of-words representations. At the same time, we expect that richer transformer architectures would enhance classification performance during fine-tuning. However, if the representation is already strong enough, is it justified to allocate additional resources for fine-tuning to achieve satisfactory results?

When the same architecture is shared for both pre-training and fine-tuning (Peters et al., 2018; Devlin et al., 2019), it becomes challenging to disentangle the relative influence of the representation and the classifier. To isolate the performance of each component, we propose an empirical study where we train both simple linear models and complex transformer models, with and without pre-trained representations, and test their performance in high and low annotation budget scenarios.

We specifically focus on investigating the previous question within the context of a low annotation budget scenario, where the availability of labeled data for fine-tuning is limited.

Our empirical study shows that:

- In low-budget scenarios, the incorporation of pre-trained representations results in a more significant performance improvement compared to high-budget scenarios. Moreover, when we isolate the gains attributed to pre-training, the performance gains of transformers over simpler models become marginal, meaning that the quality of the representations is the most important component.
- In this setting, a simple classifier on top of a contextual representation achieves competitive results compared to fine-tuning. Consequently, the impact of the classifier proves to be rather minimal, allowing us to utilize more cost-effective alternatives.

2 Related Work

While transformer (Vaswani et al., 2017) architectures are known to benefit from large amounts of training data for optimal performance (Ezen-Can, 2020; Kirstain et al., 2022), the pre-training plus fine-tuning approach has also shown promising results in low annotation budget scenarios (Ein-Dor

Proceedings of the Fifth Workshop on Insights from Negative Results in NLP, pages 17–24

et al., 2020; Tamkin et al., 2022; Shelmanov et al., 2021; Zhang et al., 2022).

Fine-tuning is thought to adjust the pre-trained representations in order to simplify the downstream task (Zhou and Srikumar, 2022, 2021). However, the fine-tuning step itself can be unstable (Mosbach et al., 2021; Zhang et al., 2021) and sensitive to weight initialization (Dodge et al., 2020). These issues are particularly pronounced in low-budget scenarios (Margatina et al., 2022). To address these challenges, researchers have explored techniques such as parameter reduction (Han et al., 2021; He et al., 2021; Liu et al., 2018) or modifications to the fine-tuning procedure (Hua et al., 2021; Yang and Ma, 2022). Other authors have explored the possibility of using pre-trained representations directly with simpler classifiers (Li et al., 2021; Dubey et al., 2018)

The importance of representation choice has lately received a significant amount of attention from the active learning (AL) community (Schröder and Niekler, 2020; Zhang et al., 2017; Ein-Dor et al., 2020; Yuan et al., 2020; Yauney and Mimno, 2021; Margatina et al., 2022; Shelmanov et al., 2021). Most of the research in AL attempts to quantify what representation is best when training the initial model for active learning, which is usually referred to as the cold start problem (Lu and MacNamee, 2020; Zhang et al., 2022). The importance of word embeddings has also been studied in the context of highly imbalanced data scenarios (Sahan et al., 2021; Naseem et al., 2021; Hashimoto et al., 2016; Kholghi et al., 2016).

The main difference between our work and previous literature is that in prior studies, the fine-tuning process involved the simultaneous updates of both the pre-trained weights and the classifier, without considering their relative importance. Having established the relevance of the representation, especially in few-shot learning scenarios, we aim to investigate whether fine-tuning complex architectures in classification tasks is justified.

3 The Role of the Classifier in Low-budget Scenarios

To conduct our study, we aim to compare the performance of a transformer-based model and a simple classifier, trained with and without pre-trained representations. The main focus of our investigation will be on scenarios with a limited annotation budget, by utilizing learning curves. Each point in these curves represents a specific training size, enabling us to evaluate the model's performance as the data size increases. Additionally, we will report performance on the full dataset for the different models. Next, we detail the models, datasets, and learning curves employed.

Dataset	Size	Prior	Len.	Task
IMDB	50K	50%	313	sentiment
WiTox	224K	9%	78	toxicity
S140	1.6M	50%	23	sentiment
CivCom	2M	8%	58	toxicity

Table 1: Datasets statistics with the number of samples, target (positive) class prior, average token sequence length, and classification task.

3.1 Models

We contrast two model architectures: a transformer (BERT) and a max-entropy model (MaxEnt). Each of the models will be trained in two settings: 1) without pre-trained representations and 2) with pre-trained representations.

BERT (Devlin et al., 2019): $\text{BERT}_{\text{BASE-uncased}}$ model (110M parameters) using standard pre-training (BooksCorpus plus Wikipedia) and implemented using the HuggingFace Transformers library (Wolf et al., 2020). Learning without pre-trained representations means learning with randomly initialized weights (similar to Voita and Titov, 2020 and Zhang and Bowman, 2018). The hyper-parameter values can be found in A.2.

MaxEnt: A standard max-entropy model trained with l_2 regularization. When training without pre-trained representations, we used a sparse bag-of-n-grams representation. For the models with pre-training, we extracted static representations from the second-to-last hidden layer (Bommasani et al., 2020; Devlin et al., 2019) using the average of BERT's token embeddings (768 dimensions vectors). Our preliminary experiments have shown that such embeddings yield better performance than using BERT's `[CLS]` token (similar to the ablation studies in Devlin et al., 2019 and the observations presented in Lu and MacNamee, 2020). The regularization parameters and the optimal n-gram size were validated via 5-fold cross-validation.

3.2 Datasets

We use four textual classification datasets with both balanced and imbalanced label distributions,

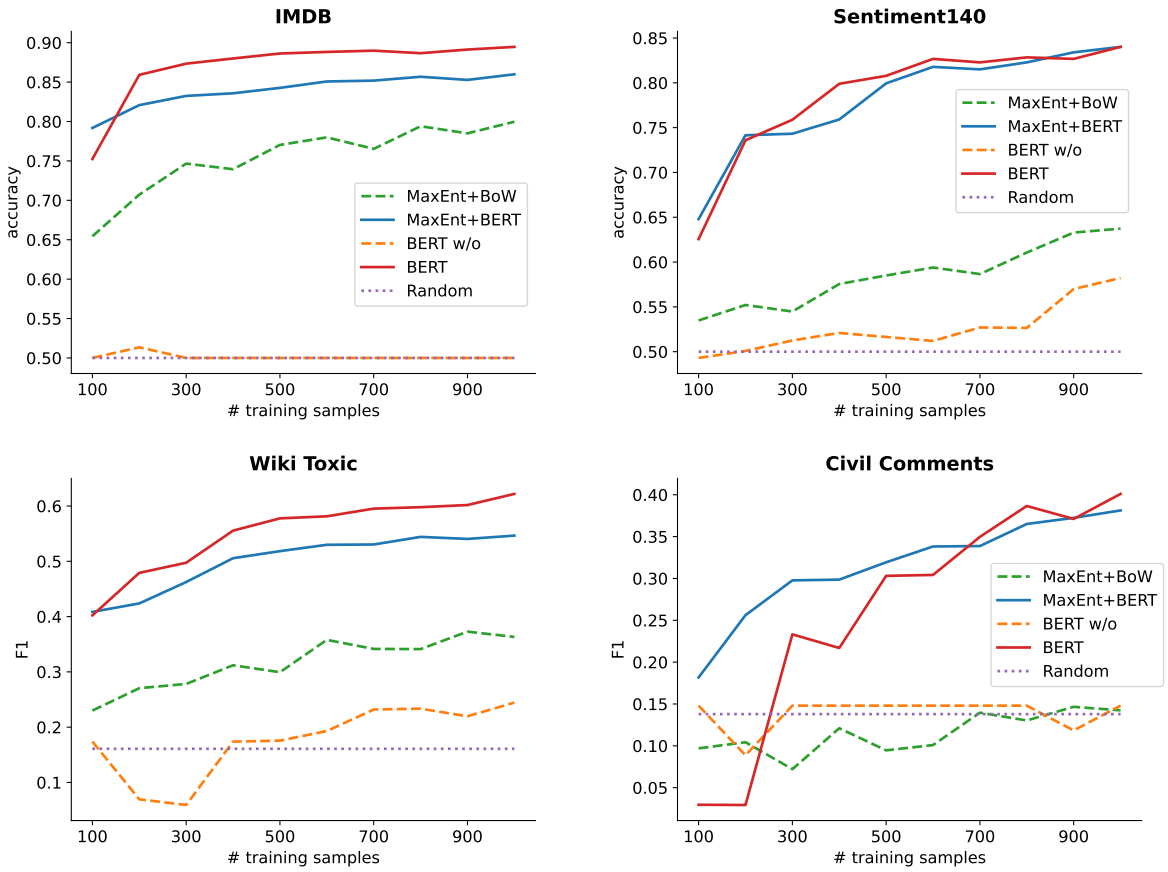

Figure 1: Performance of different models when learning with a limited annotation budget on various datasets. 'w/o' means without pre-trained representations. We also report the expected performance of a random classifier predicting i.i.d. labels.

encompassing two significant classification tasks (sentiment analysis and toxicity detection) across a variety of language registers and input lengths:

IMDB (Maas et al., 2011): Movie reviews annotated with sentiment labels. This is a dataset with a balanced distribution of labels.

Wikipedia Toxicity (WiTox; Wulczyn et al., 2017): Wikipedia discussion comments annotated with toxicity labels. This is a dataset with a highly imbalanced label distribution: less than 10% of the labels correspond to toxic comments.

Sentiment140 (S140; Go et al., 2009): A balanced dataset of Twitter messages annotated with sentiment.

Civil Comments (CivCom; Borkan et al., 2019): Opinions posted in the Civil Comments platform annotated for toxic behavior. This dataset exhibits a significantly skewed label distribution, with less than 10% toxic comments.

For Wikipedia Toxicity and Civil Comments, we have applied a pre-processing consisting of removing all markup code and non-alpha-numeric characters except relevant punctuation. Table 1 presents the datasets' summary statistics.

3.3 Learning curves

For our study, we generate learning curves where each point corresponds to a different training size with a budget of N samples. We create training sets by selecting the N random samples incrementally. N ranges from 100 to 1000 in increments of 100. At each step, new samples are added to the existing selection.

For every model, some hyper-parameters need optimization. At every point N in the learning curve, we create an 80/20% 5-fold cross-validation split and validate the optimal hyper-parameters. We then use these hyper-parameters to train a model using all the N training samples, and its performance is evaluated on the test set.

We repeat the experiments using 5 training sets and initializing the parameters using different random seeds. We report the mean results. As evaluation metrics we use: accuracy for the balanced

Dataset	Model	ALC	
		w/o	p.r.
IMDB	MaxEnt	0.75	0.84
	BERT	0.50	**0.87**
WiTox	MaxEnt	0.32	0.50
	BERT	0.18	**0.51**
CivCom	MaxEnt	0.11	**0.32**
	BERT	0.15	0.26
S140	MaxEnt	0.58	**0.79**
	BERT	0.53	**0.79**

Table 2: Model performance with a limited annotation budget, using pre-trained representations (p.r.) and without (w/o). We report the area under the learning curve (ALC) from 100 to 1000 examples, using accuracy for balanced datasets and F1 (of the target class) for imbalanced datasets. The best model performance for each dataset is reported in bold.

Dataset	Model	Performance	
		w/o	p.r.
IMDB	MaxEnt	0.89	0.89
	BERT	0.53	**0.93**
	Random	0.50	0.50
WiTox	MaxEnt	0.66	0.61
	BERT	0.48	**0.68**
	Random	0.16	0.16
CivCom	MaxEnt	0.60	0.57
	BERT	0.15	**0.70**
	Random	0.14	0.14
S140	MaxEnt	0.81	**0.86**
	BERT	0.77	**0.86**
	Random	0.50	0.50

Table 3: Model performance using all training data.

datasets (IMDB and Sentiment140) and F1 (of the target class) for the imbalanced datasets (Wikipedia Toxicity and Civil Comments).

In total, we performed 400 experiments for each model: 4 datasets, with and without pre-trained representations, 5 seeds, and 10 learning points. For BERT, the computation of each learning point took 27 minutes on average on a single Nvidia V100 GPU, totaling 177 hours of GPU computation. A.1 contains further details about the running times.

4 Results

Figure 1 shows our main results on analyzing the performance of models under the low-budget annotation setting. To summarize the learning curve results, we also compute a single performance score for each model: the area under the learning curve (ALC). This provides us with a more robust metric to compare the different models for a dataset[1]. Table 2 shows the results obtained.

We observe that in the low-budget scenario when pre-trained representations are used, the choice of model seems to be of little importance. Both the complex transformer model and a simple linear max-entropy model perform similarly.

In addition, when only very few labels are available (first curve points in Figure 1), the simpler model seems to outperform the more complex one. MaxEnt demonstrates a more stable behavior within this range, due to its fewer number of parameters. This shows that when the training set is small there is not much to be gained from fine-tuning all the layers of the model.

The biggest difference in performance in the low-budget scenario comes from the representation and not the architecture. In fact, without pre-trained representations, the more complex models perform significantly worse than simpler models. Pre-trained representations seem to be capturing some properties of the input space that can be exploited by all models. We suppose that since pre-training implicitly induces a distance space over words, models using pre-trained representations generalize more easily to unseen words. This would explain why pre-trained representations are especially helpful in the low-annotation budget scenario since generalization to unseen words is critical in this case.

Table 3 presents the performance results obtained by employing the entire training set. Within this data-rich scenario, typically used for model comparison, we first confirm the well-established fact that BERT with pre-trained weights yields better results than simpler models (Devlin et al., 2019). Interestingly, in this context, simpler models do not seem to obtain significant benefits from the use of pre-trained representations. Unlike the low-budget scenario, in this setting, fine-tuning all layers of the model results in significant performance improvements.

[1]Direct performance comparison across datasets is not always feasible because the underlying score may vary.

5 Conclusion

In this paper, we studied classifiers in a low-budget scenario, analyzing the impact of fine-tuning on performance by separating the benefits derived from pre-training weights from those of architectural fine-tuning.

Based on our findings, we recommend testing simple models that incorporate pre-trained representations before investing resources in fine-tuning complex models. In fact, when labeled data is scarce, the role of the representations is crucial, and the use of pre-trained representations enhances performance across all models, regardless of their complexity. As a result, the choice of classifier becomes irrelevant in this context compared to the quality of the representations. The marginal performance gains offered by more sophisticated architectures may not justify the additional computational resource demands.

Limitations

When studying the performance of a simple classifier over pre-trained representations, we have considered BERT as the representative for transformer-based models. A comparison with other transformer models, with a different number of parameters and embedding representations, would make our conclusions more general.

Our analysis is limited to binary classification tasks. Future research should aim to extend our study to other types of tasks to better understand the broader implications of our findings.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 853459. The authors gratefully acknowledge the computer resources at ARTEMISA, funded by the European Union ERDF and Comunitat Valenciana as well as the technical support provided by the Instituto de Física Corpuscular, IFIC (CSIC-UV). This research is supported by a recognition 2021SGR-Cat (01266 LQMC) from AGAUR (Generalitat de Catalunya).

References

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. *arXiv:1903.04561 [cs, stat]*. ArXiv: 1903.04561.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. 2018. Maximum-entropy fine grained classification. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Aysu Ezen-Can. 2020. A comparison of LSTM and BERT for small corpus. *CoRR*, abs/2009.05451.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification using Distant Supervision. *CS224N project report, Stanford*, 1(12):6.

Wenjuan Han, Bo Pang, and Ying Nian Wu. 2021. Robust transfer learning with pretrained language models through adapters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 854–861, Online. Association for Computational Linguistics.

Kazuma Hashimoto, Georgios Kontonatsios, Makoto Miwa, and Sophia Ananiadou. 2016. Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of Biomedical Informatics*, 62:59–65.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based

tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online. Association for Computational Linguistics.

Hang Hua, Xingjian Li, Dejing Dou, Chengzhong Xu, and Jiebo Luo. 2021. Noise stability regularization for improving BERT fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3229–3241, Online. Association for Computational Linguistics.

Mahnoosh Kholghi, Lance De Vine, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. 2016. The Benefits of Word Embeddings Features for Active Learning in Clinical Information Extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 25–34, Melbourne, Australia.

Yuval Kirstain, Patrick Lewis, Sebastian Riedel, and Omer Levy. 2022. A few more examples may be worth billions of parameters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1017–1029, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Linyang Li, Demin Song, Ruotian Ma, Xipeng Qiu, and Xuanjing Huang. 2021. Knn-bert: Fine-tuning pre-trained models with knn classifier.

Liyuan Liu, Xiang Ren, Jingbo Shang, Xiaotao Gu, Jian Peng, and Jiawei Han. 2018. Efficient contextualized representation: Language model pruning for sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1215–1225, Brussels, Belgium. Association for Computational Linguistics.

Jinghui Lu and Brian MacNamee. 2020. Investigating the effectiveness of representations based on pretrained transformer-based language models in active learning for labelling text datasets. *arXiv preprint arXiv:2004.13138*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 825–836, Dublin, Ireland. Association for Computational Linguistics.

Alessio Miaschi and Felice Dell'Orletta. 2020. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119, Online. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Usman Naseem, Matloob Khushi, Shah Khalid Khan, Kamran Shaukat, and Mohammad Ali Moni. 2021. A Comparative Analysis of Active Learning for Biomedical Text Mining. *Applied System Innovation*, 4(1):23.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training.

Marko Sahan, Vaclav Smidl, and Radek Marik. 2021. Active Learning for Text Classification and Fake News Detection. In *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, pages 87–94. IEEE Computer Society.

Christopher Schröder and Andreas Niekler. 2020. A Survey of Active Learning for Text Classification using Deep Neural Networks. ArXiv:2008.07267 [cs] version: 1.

Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. Active learning for sequence tagging with deep pre-trained models and Bayesian uncertainty estimates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1698–1712, Online. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Alex Tamkin, Dat Nguyen, Salil Deshpande, Jesse Mu, and Noah Goodman. 2022. Active learning helps pretrained models learn the intended task.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex Machina: Personal Attacks Seen at Scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Patrick Xia, Shijie Wu, and Benjamin Van Durme. 2020. Which *BERT? A survey organizing contextualized encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7516–7533, Online. Association for Computational Linguistics.

Chenghao Yang and Xuezhe Ma. 2022. Improving stability of fine-tuning pretrained language models via component-wise gradient norm clipping. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4854–4859, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gregory Yauney and David Mimno. 2021. Comparing text representations: A theory-driven approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5527–5539, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*.

Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active Discriminative Text Representation Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2021. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Online. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 BERT Runtime

Table 4 shows BERT's training plus testing running times for the budgets considered in the learning curves studied in this work. These experiments were performed using a single Nvidia V100 GPU.

Budget	IMDB	WiTox	CivCom	S140
100	15:07	38:02	27:30	01:00
200	17:24	40:17	27:31	01:21
300	19:57	41:54	27:45	01:45
400	21:35	42:27	29:57	02:18
500	22:54	43:02	31:35	03:03
600	25:19	47:09	32:28	03:24
700	26:50	44:50	34:02	03:35
800	28:48	48:09	34:59	03:40
900	31:02	48:40	35:45	04:01
1000	30:35	56:12	36:20	04:15

Table 4: BERT training and testing average runtime.

Table 5 displays the average speed of embedding generation, measured in samples per second.

Dataset	Gen. Time
IMDB	37.48 i/s
Sentiment140	135.42 i/s
Wiki Toxic	186.20 i/s
Civil Comments	131.37 i/s

Table 5: Embedding generation average speed.

Compared to fine-tuning, embedding extraction is a significantly more efficient operation and can feasibly be computed on the CPU.

A.2 Experimental Details

Table 6 contains a summary of BERT hyper-parameters used in the experiments.

Hyper-parameter	Value
Max. training epochs	10
Learning rate	$5 \cdot 10^{-5}$
AdamW λ	0.0
AdamW β_1	0.9
AdamW β_2	0.999
Attention dropout	0.1
Hidden dropout	0.1
Mixed Precision	fp16
Seq. length (IMDB)	350
Seq. length (Wiki Toxic)	150
Seq. length (Civil Comments)	150
Seq. length (Sentiment140)	50
Batch size (IMDB)	20
Batch size (Wiki Toxic)	50
Batch size (Civil Comments)	50
Batch size (Sentiment140)	64

Table 6: BERT hyper-parameters.