Are ELECTRA's Sentence Embeddings Beyond Repair? The Case of Semantic Textual Similarity

Ivan Rep David Dukić Jan Šnajder
TakeLab, Faculty of Electrical Engineering and Computing, University of Zagreb
irep2718@gmail.com
{david.dukic, jan.snajder}@fer.hr

Abstract

While BERT produces high-quality sentence embeddings, its pre-training computational cost is a significant drawback. In contrast, ELECTRA provides a cost-effective pre-training objective and downstream task performance improvements, but worse sentence embeddings. The community tacitly stopped utilizing ELECTRA's sentence embeddings for semantic textual similarity (STS). We notice a significant drop in performance for the ELECTRA discriminator's last layer in comparison to prior layers. We explore this drop and propose a way to repair the embeddings using a novel truncated model fine-tuning (TMFT) method. TMFT improves the Spearman correlation coefficient by over 8 points while increasing parameter efficiency on the STS Benchmark. We extend our analysis to various model sizes, languages, and two other tasks. Further, we discover the surprising efficacy of ELECTRA's generator model, which performs on par with BERT, using significantly fewer parameters and a substantially smaller embedding size. Finally, we observe boosts by combining TMFT with word similarity or domain adaptive pre-training.

1 Introduction

Pre-trained language models (PLMs) have been a staple in NLP for years, leveraging self-supervised objectives to improve representations for downstream tasks. BERT (Devlin et al., 2019), one of the most widely used PLMs, uses a masked language modeling (MLM) objective for pre-training. The main drawbacks of MLM are the substantial compute cost due to the low masking rate, and the gap between the pre-training task and downstream tasks. To address these issues, Clark et al. (2020) introduce ELECTRA, which substitutes MLM with replaced token detection (RTD), achieving the same results as BERT but with four times less compute. RTD uses a generator model that corrupts the input, while a discriminator model distinguishes between corrupted and original tokens.

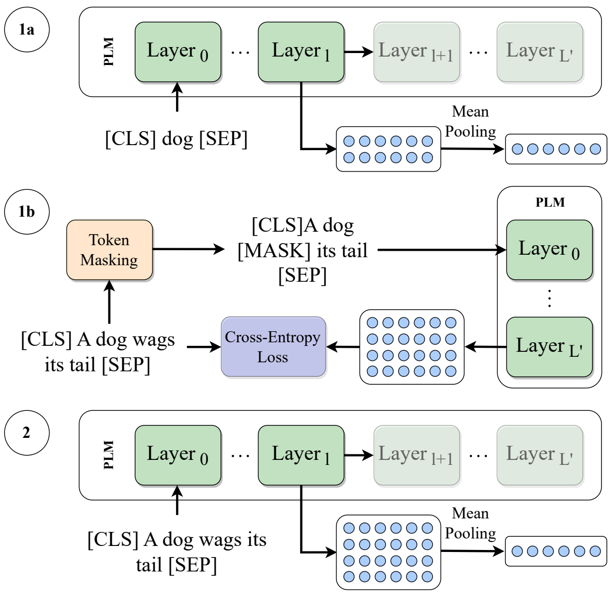


Figure 1: A method for improving sentence embeddings with (2) TMFT on STS. We apply mean pooling over the embeddings at layer l and fine-tune. One of the combinations can also be added for further improvement: (1a) TMFT on word similarity, or (1b) DAPT using MLM.

Semantic textual similarity (STS) (Agirre et al., 2013) is a foundational NLP task with broad applications. STS applications must balance accuracy and embedding size to ensure fast inference. To this end, Reimers and Gurevych (2019) introduced sentence transformers, a framework based on the transformer bi-encoder architecture. It encodes text representations independently before pooling them, calculating the similarity score through a comparison operation between the two embedded texts.

While some transformer models excel across GLUE (Wang et al., 2018) tasks, their sentence embedding quality may vary. Sentence transformers often use PLMs pre-trained with a language modeling objective (Song et al., 2020; Raffel et al., 2020; Liu et al., 2019). In contrast, RTD pre-trained models demonstrate subpar performance in a bi-encoder setting (Reimers, 2021), but pre-training such a model is more cost-effective. It is unclear whether this low performance is due to pre-training and whether the embeddings can be improved.

To address this gap, we present a layer-wise study of ELECTRA's sentence embeddings. We examine the performance discrepancy between BERT and ELECTRA's discriminator across various model sizes, languages, and tasks. We hypothesize the last few layers of the discriminator are too specialized for the pre-training task. Following this, we present truncated model fine-tuning (TMFT), shown in Figure 1, which utilizes the transformer up to layer l, followed by pooling and fine-tuning the whole truncated model. We show that the discriminator suffers from a drop in STS performance when fine-tuning the final layer's embeddings and that this drop is consistent across all inspected model sizes and languages. Although we focus on STS, we expose a similar trend for two other tasks. When applying TMFT on STS Benchmark (STSB) (Cer et al., 2017), the embeddings significantly outperform the ones from the ELECTRA discriminator's last layer. Moreover, TMFT-obtained embeddings outperform the ones of BERT fine-tuned on STSB up to but not including the eighth layer. Further, we uncover the efficacy of the ELECTRA generator model, performing on par with BERT while having a significantly smaller embedding size and substantially fewer parameters. Finally, we propose two improvements of the basic TMFT method: prior TMFT on word similarity and prior domain adaptive pre-training (DAPT) using MLM, shown in Figure 1.

Our contributions are: (1) a layer-wise analysis of ELECTRA's sentence embeddings for various model sizes, languages, and tasks; (2) a novel TMFT method for repairing ELECTRA's embeddings that substantially improves performance on STS, paraphrase identification, and textual entailment tasks, and exposing the surprising efficacy of the generator model; (3) two additional techniques in combination with TMFT for improving ELECTRA's embeddings even further on STS. [1]

2 Related Work

Reimers and Gurevych (2019) introduced Siamese networks to transformers, motivating significant research on enhancing sentence embeddings. Fine-tuning on an auxiliary task unrelated to STS has also been explored. Reimers and Gurevych (2019) first fine-tune on a textual entailment task, followed by fine-tuning on STS. DAPT has also become a widely adopted method, improving performance in downstream tasks (Gururangan et al., 2020).

Similarly, there has been considerable interest in using different layers of a PLM for sentence embeddings. Bommasani et al. (2020) assess the layer-wise performance of transformer models and pooling methods on word similarity datasets. Huang et al. (2021) determine what combinations of hidden states perform the best for unsupervised STS, while Jawahar et al. (2019) extract layer-wise structural characteristics encoded in BERT by probing. Finally, Ethayarajh (2019) explores layer-wise embedding anisotropy. Two works that resemble ours the most are Hosseini et al. (2023) and Li et al. (2024). The former improves results by combining layer representations with dynamic programming, while we fine-tune the model from input embeddings up to a specific layer. Li et al. (2024), developed concurrently with our work, uses model and embedding truncation combined with fine-tuning and a novel loss function. However, this work does not address ELECTRA's performance drop.

3 Truncated Model Fine-Tuning

The usual approach for obtaining sentence embeddings is applying a pooling operation over the last layer's embeddings. We use mean pooling as it yields the best results, in line with previous work (Reimers and Gurevych, 2019). The TMFT method we propose for repairing ELECTRA's embeddings reduces to taking the l-th layer output followed by pooling and fine-tuning on the target task.

For a sentence $S = (s_1, \ldots, s_N)$ the encoder outputs a tensor $E \in \mathbb{R}^{L' \times N \times d}$, where L' is the number of layers including the input embeddings (which we treat as a layer). We then apply mean pooling p over the l-th representation, $p(E, l) = \frac{1}{N} \sum_{n=1}^{N} E_{l,n,:}$, where $E_{l,n,:}$ is the d-dimensional output token embedding of layer l for token at position n. We apply this to both sentences, compare them using cosine similarity, and propagate the loss from layer l to the model input.[2]

Furthermore, we propose combining one supervised and one self-supervised method with TMFT

[1]We provide the source code for our work here: https://github.com/ir2718/similarity-embedding-quality

[2]Fine-tuning based on the final embedding and using prior embeddings for inference yields unsatisfactory results, and a similar approach is already known (Hosseini et al., 2023).

for performance gains: (1) PLM fine-tuning on the word similarity task or (2) DAPT using MLM. The intuition for the former is that word similarity is crucial for assessing sentence similarity, while the latter builds upon prior evidence that MLM-based PLMs work well in a bi-encoder (Reimers, 2021).

4 Experiments and Results

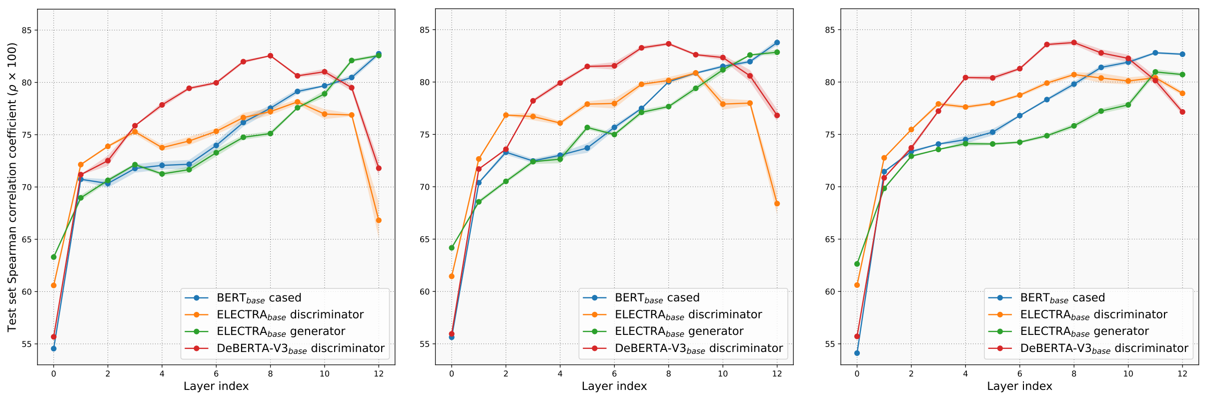
We fine-tune each model on the STSB from GLUE. We also run the experiments on machine-translated versions of STSB in Korean, German, and Spanish (cf. Appendix B). We choose these based on language-specific dataset and PLM availability. We report the Spearman correlation coefficient, suggested by Reimers and Gurevych (2019). For word similarity experiments, we use word pairs present in the following datasets: RG-65 (Rubenstein and Goodenough, 1965), WordSim-353 (Finkelstein et al., 2001), SimLex-999 (Hill et al., 2015), and SimVerb-3500 (Gerz et al., 2016), with a random 70:15:15 train, validation, and test split. The MLM experiments are conducted on sentences from SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). We also experiment with paraphrase identification and textual entailment using the Microsoft Research Paraphrase Corpus (MRPC) and Sentences Involving Compositional Knowledge (SICK) textual entailment dataset (Marelli et al., 2014), respectively (cf. Appendix A). For STSB, MRPC, and SICK, we use cross-validation splits defined by the dataset authors.

We use the following models from HuggingFace Transformers (Wolf et al., 2020): BERT tiny, mini, small, medium, base, and large. For the ELECTRA discriminator and generator, we use the small, base, and large models. To strengthen our findings on RTD-pre-trained models, we run experiments for the DeBERTaV3 model (He et al., 2021). For fine-tuning and pre-training, we use the AdamW optimizer (Loshchilov and Hutter, 2017), a learning rate of $2e{-}5$, and weight decay set to $1e{-}2$. We apply gradient clipping to a max norm of 1.0. For fine-tuning on all tasks except word similarity, we use a batch size of 32 for 10 epochs. Word similarity fine-tuning uses a batch size of 128 for 50 epochs. For DAPT, we use a batch size of 32 with 8 gradient accumulation steps for 10 epochs with 0.15 masking probability. An exception is DeBERTaV3, with a batch size of 8 and 32 gradient accumulation steps due to memory constraints. All reported results are averaged across five seeds. Test set Spearman correlation coefficients on STSB correspond to the model with the highest Spearman correlation coefficient on the validation set. For MRPC and SICK, the test set F1 scores correspond to the models with the classification threshold optimized for the highest validation set F1 score.

4.1 ELECTRA

Applying ELECTRA to downstream tasks is usually done using the discriminator. We decided not to follow this practice, as ELECTRA's authors do not give convincing reasons for discarding the generator. Hence, we conduct experiments with both models. The generator is similar to BERT, except the generator's input embeddings are tied to the discriminator in pre-training. For comparison, we use BERT as a baseline as it is a standard choice and similar in size to the discriminator.

Figure 2a shows test set Spearman correlation coefficients for TMFT applied to ELECTRA$_{\text{base}}$ discriminator, ELECTRA$_{\text{base}}$ generator, and BERT$_{\text{base}}$. BERT shows a trend where the Spearman correlation coefficient roughly increases as the index of the fine-tuned layer embedding increases. The same trend is present for the generator. ELECTRA$_{\text{base}}$ generator with 33.31M parameters maintains comparable performance to BERT$_{\text{base}}$ with 107.72M parameters on all tasks (cf. Table 1, Figure 5, and Figure 6 in Appendix A). This finding is consistent across all inspected generator and BERT sizes (cf. Table 2 in Appendix D). The discriminator shows a different trend, gradually increasing until the ninth layer, after which performance drops sharply. We attribute this drop to the RTD task, also suggested by Centered Kernel Alignment (CKA) representation similarity analysis, which shows that CKA values between discriminator models and MLMs drop in the final layers even before fine-tuning (cf. Figure 3). A similar performance drop occurs for Korean, German, and Spanish (cf. Appendix B), all considered discriminator model sizes (cf. Figure 11), the paraphrase identification task (cf. Figure 5), and the DeBERTaV3 discriminator (cf. Figure 2a). The increase in the ELECTRA discriminator test Spearman correlation coefficients between the ninth (87.63M parameters) and last state (108.89M parameters) is 11.32 points. The best-performing state on the validation set is the third state (45.10M parameters), and the increase for the third state compared to the last state on the test set is 8.36. Furthermore, the discriminator's input embeddings outperform BERT, while its output



(a) TMFT on STSB

(b) TMFT on word similarity datasets + TMFT on STSB

(c) MLM on NLI datasets + TMFT on STSB

Figure 2: Test set Spearman correlation coefficients on STSB using TMFT with and without improvements (shaded area is the standard deviation). Subfigure 2a presents results using TMFT on STSB, 2b shows TMFT on STSB with prior TMFT on word similarity, and 2c depicts TMFT on STSB with prior MLM. More details are in Table 1.

	TMFT STSB				**TMFT WS + TMFT STSB**				**DAPT NLI + TMFT STSB**			
Model	**Layer**	**Params**	**Val**	**Test**	**Layer**	**Params**	**Val**	**Test**	**Layer**	**Params**	**Val**	**Test**
BERT$_{base}$	12	107.72M	86.07/85.98	**82.74/83.03**	12	107.72M	85.85/85.84	**83.77/84.08**	12	107.72M	85.91/85.71	82.66/82.64
ELECTRA$_{D\ base}$	3	45.10M	82.15/82.20	75.29/76.96	3	45.10M	82.66/82.56	76.71/77.31	7	73.45M	84.07/83.78	79.90/79.92
ELECTRA$_{G\ base}$	12	33.31M	**86.62/86.38**	82.57/82.50	12	33.31M	**86.67/86.39**	82.85/82.91	11	32.52M	85.58/85.22	80.97/80.85
DeBERTaV3$_{base}$	7	148.00M	84.80/84.86	81.98/82.44	7	148.00M	85.48/85.55	83.27/83.31	7	148.00M	**85.95/85.88**	**83.59/83.51**

Table 1: Comparison of models with the highest validation set Spearman correlation coefficient using TMFT. The reported scores are the test set Spearman and Pearson correlation coefficients. Results with the addition of improvements are included (WS stands for word similarity, NLI stands for natural language inference). Bold values represent the highest values for the used method across all trained models.

embeddings surpass BERT's up to, but not including, the eighth layer. Across all models, the largest increase between consecutive layers is between the 0th and 1st layers, likely due to self-attention.

4.2 Further Improvements

Our first proposed improvement is TMFT on word similarity before TMFT on sentence similarity. The downside of this method is that it requires data labeled for word similarity. We consider only using the same layer for both fine-tuning procedures. Figure 2b gives the test set Spearman correlation coefficients on STSB for the proposed improvement. We observe BERT and the discriminator both benefit from this method, with ELECTRA outperforming BERT up to but not including the tenth layer.

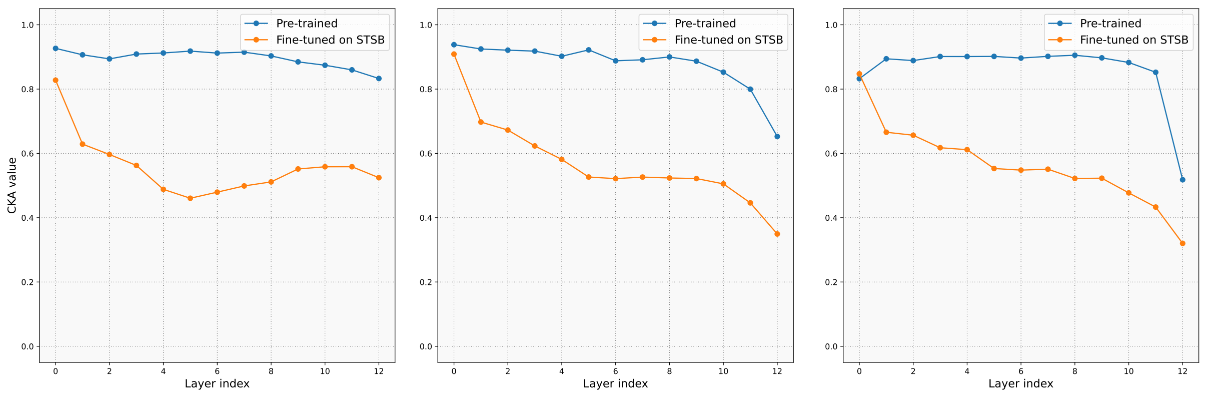
Our second improvement is DAPT using MLM before TMFT on STS, regardless of the model type. We only consider pre-training using the last layer's output. Figure 2c presents the test set Spearman correlation coefficients on the STSB dataset. With improvements included, the performance of almost all representations improved across all models, and the drop in the final layers of the discriminator is diminished (cf. Figure 2c and Table 1).

4.3 Parameter-Performance Trade-off

Finally, we investigate the parameter-performance trade-off. The results show the number of parameters the model has when using the best-performing representation on the validation set and the corresponding test set Spearman correlation coefficients (cf. Figure 4 and Table 2). We consider only TMFT without improvements. Considering the test set scores and the number of parameters, the best models are BERT (tiny, mini, large) and ELECTRA generator (small, base, large). The difference in the number of parameters and layers suggests the depth of a transformer is essential for retaining performance on STS. Figure 4 also demonstrates an improvement in the number of parameters, and in the performance when comparing the last and best representation of the discriminator.

4.4 Performance Drop Analysis

To explain the drop in the Spearman correlation coefficient, we conduct an analysis using CKA (Kornblith et al., 2019). We apply the CKA to the



(a) ELECTRA$_{\text{G base}}$ and BERT$_{\text{base}}$ (b) ELECTRA$_{\text{D base}}$ and BERT$_{\text{base}}$ (c) DeBERTaV3$_{\text{D base}}$ and BERT$_{\text{base}}$

Figure 3: The result of applying CKA on the hidden layer representations of the STSB test set at a layer with a certain index. Subfigure 3a presents the comparison between ELECTRA generator and BERT, subfigure 3b the comparison between ELECTRA discriminator and BERT, and subfigure 3c the comparison between DeBERTaV3 discriminator and BERT.

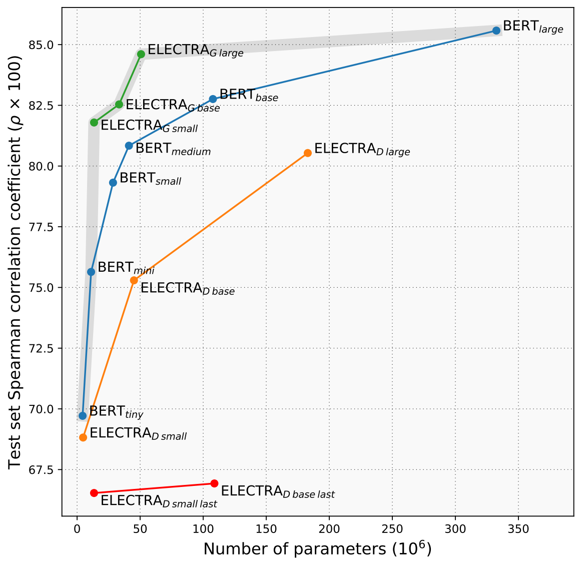


Figure 4: Comparison of the number of parameters of the model and the test set Spearman correlation coefficients. The shown models have the highest validation Spearman correlation coefficient value. The figure also includes the last layer representations that do not correspond to the highest validation Spearman correlation coefficient. ELECTRA$_{\text{large}}$ discriminator is excluded as its value is too small (25.84). The gray line indicates the Pareto front. For detailed test set Spearman correlation coefficient values, refer to Table 2 in Appendix D.

STSB test set representations obtained from the hidden states of two models with the same index. Since the time complexity of the TMFT method depends on the dataset size and the number of hidden states, it is beneficial to know about the presence of the possible drop prior to fine-tuning. An additional benefit of this method is that it works with pre-trained models, saving time and resources. We observe a pattern where the drop in CKA values in the final layers is a necessary condition for the drop in Spearman correlation coefficient values in the final layers. The comparison between the ELECTRA discriminator and BERT (Figure 3b) exhibits the same trend as the comparison between the DeBERTaV3 discriminator and BERT (Figure 3c), which is a sharp decline in CKA values in the final few layers. The comparison between the ELECTRA generator and BERT (Figure 3a) does not exhibit the CKA value drop. All combinations of inspected models after fine-tuning show similar trends compared to their pre-trained counterparts.The drop could be attributed to the model architecture, pre-training data, pre-training method, fine-tuning data, or hyperparameter choice. To control for these confounders, we apply TMFT to randomly initialized models. We found that all models exhibit the same behavior, suggesting the architecture and tokenizer choice are not to blame for the performance drop (cf. Figure 13 in Appendix E).

5 Conclusion

We analyze ELECTRA's sentence embeddings for STS and two other tasks, comparing them to a BERT baseline on the STSB dataset. Our proposed truncated model fine-tuning method significantly improves the discriminator. The generator model matches BERT's performance while significantly reducing the number of parameters and producing smaller embeddings.

6 Limitations

Our study reveals substantial differences in model performance on some tasks, but sentence embedding models can also be used for information retrieval. We limited ourselves to one dataset for STS, paraphrase identification, and entailment, four datasets for word similarity, and two datasets for MLM, due to computation restrictions. Experiments in Korean, German, and Spanish were exclusively conducted using the truncated model fine-tuning method. Another limitation of our work is the way we apply word similarity fine-tuning prior to truncated model fine-tuning on STS. We only consider fine-tuning the same layer's embeddings for both procedures. Furthermore, in MLM, prior to truncated model fine-tuning on STS, we only use the last layer's embeddings, as opposed to using previous layers. For all our experiments, we fixed the hyperparameters. Finally, our study covers only two families of models in detail: BERT and ELECTRA.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (*SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

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.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. volume 20, pages 406–414.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. KorNLI and KorSTS: New benchmark datasets for Korean natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 422–430, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

MohammadSaleh Hosseini, Munawara Munia, and Latifur Khan. 2023. BERT has more to offer: BERT layers combination yields better sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15419–15431, Singapore. Association for Computational Linguistics.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.

Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2024. 2d matryoshka sentence embeddings. *arXiv preprint arXiv:2402.14776*.

Yinhan Liu, 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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Philip May. 2021. Machine translated multilingual sts benchmark dataset.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers. 2021. Training state-of-the-art sentence embeddings models. Hugging Face JAX / Flax Community event to train sentence embedding models on 1B+ train examples.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. 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.

A Performance on Additional Tasks

Figure 5 represents the F1 test scores on the MRPC dataset. We use a bi-encoder optimized using a binary cross entropy loss. All other fine-tuning hyperparameters are the same as in the case of STS. ELECTRA discriminator and DeBERTaV3 show the same performance drop in the final layers. ELECTRA achieves an F1 score of 80.66 for the final hidden state, while the F1 for the ninth hidden state is 82.57. For DeBERTaV3 the difference is even greater. It achieves an F1 score of 80.69 for the final hidden state, while the tenth hidden state achieves 85.53. BERT and ELECTRA generator do not show a similar trend.

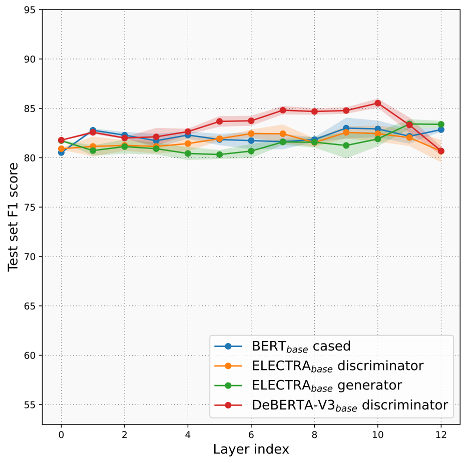


Figure 5: Test set F1 scores on the MRPC dataset across all layers.

Figure 6 represents the F1 test scores on the SICK entailment dataset. We use a bi-encoder to encode the first sentence u, the second sentence v, and concatenate its absolute difference $|u - v|$ following (Reimers and Gurevych, 2019). Finally, a classification head is used to calculate the output probabilities. The model is optimized using a cross entropy loss. All other fine-tuning hyperparameters are the same as in the case of STS. The results show slight performance drops for ELECTRA discriminator in the final layers, while DeBERTaV3 shows an upward trend in the final layers. BERT shows a slight performance drop in the final layer, while the ELECTRA generator exhibits a very large drop. The F1 value for the final layer is 65.66, while the previous layer achieves 71.87.

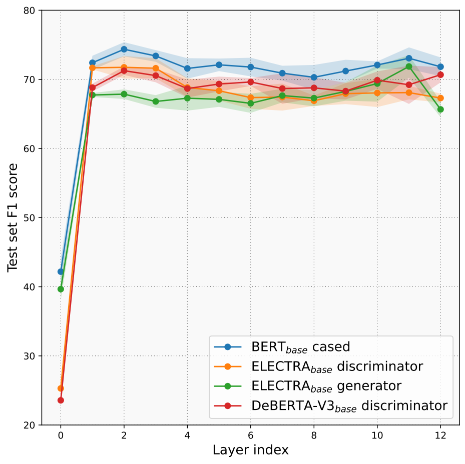


Figure 6: Test set F1 scores on the SICK entailment dataset across all layers.

B Performance on STSB in Different Languages

To strengthen our findings, we provide experiments on three other languages: Korean, German, and Spanish. Due to the scarcity of pre-trained models and labeled data, we opt for these languages. All datasets used for the experiments were machine translated. The Korean STS (Ham et al., 2020) uses an internal neural machine translation engine for all training splits, although the validation set and test set were checked for errors by annotators. The other datasets used for training are completely automatically translated using the DeepL API (May, 2021). The results for Korean, German, and Spanish are shown in Figures 7, 8, and 9 respectively.

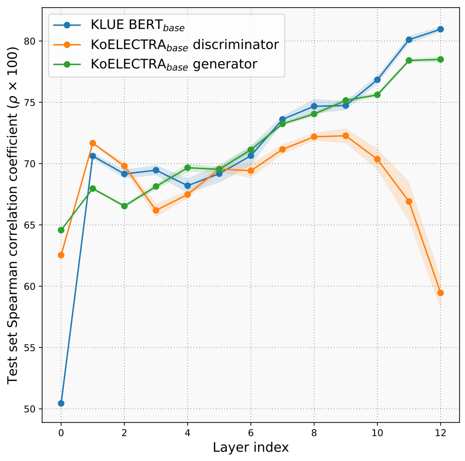


Figure 7: Test set Spearman correlation coefficients on Korean STSB across all layers.

The experiments for Korean (Figure 7) show a performance drop for the KoELECTRA discriminator.[3] The final hidden state shows a Spearman correlation coefficient of 59.46, while the ninth hidden state achieves 72.27. The KoELECTRA generator[4] achieves a Spearman correlation coefficient of 78.50 for the final hidden state, while for KLUE BERT[5] it is 80.96.

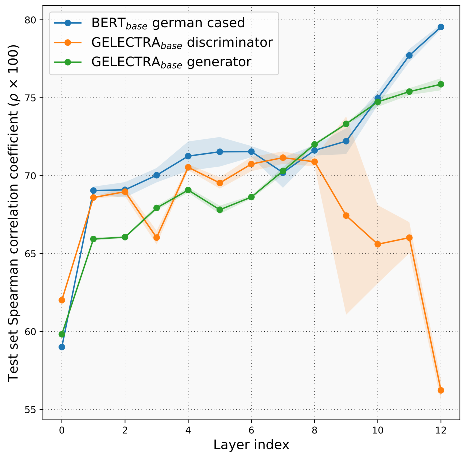


Figure 8: Test set Spearman correlation coefficients on German STSB across all layers.

Experiments in German (Figure 8) exhibit a similar trend as in Korean and English. The GELECTRA discriminator[6] achieves a Spearman correlation coefficient of 56.22, while for the seventh hidden state it is 71.15. The difference between the GELECTRA generator[7] and german BERT[8] is more apparent, with the highest Spearman correlation coefficients of 75.86, and 79.54, respectively.

Finally, the experiments in Spanish (Figure 9) are in line with the other languages as well. The Electricidad discriminator[9] achieves a Spearman correlation coefficient of 61.73 for the final hidden state. The highest value is 72.88, which is achieved for the eight hidden state. A performance gap is present for the Electricidad generator[10] and BETO model,[11] scoring 77.51, and 81.57, respectively. To summarize, experiments in all languages exhibit a performance drop for the discriminator in the final layers.

[3]https://huggingface.co/monologg/koelectra-base-v3-discriminator

[4]https://huggingface.co/monologg/koelectra-base-v3-generator

[5]https://huggingface.co/klue/bert-base

[6]https://huggingface.co/deepset/gelectra-base

[7]https://huggingface.co/deepset/gelectra-base-generator

[8]https://huggingface.co/google-bert/bert-base-german-cased

[9]https://huggingface.co/mrm8488/electricidad-base-discriminator

[10]https://huggingface.co/mrm8488/electricidad-base-generator

[11]https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased

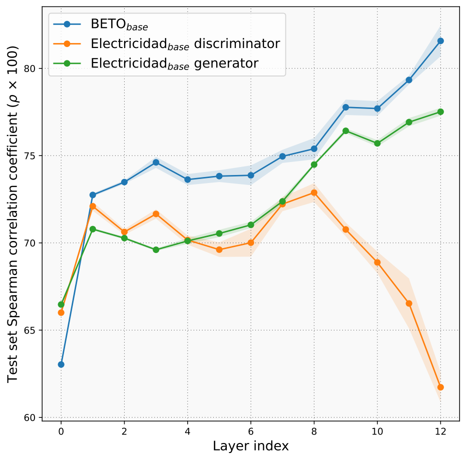


Figure 9: Test set Spearman correlation coefficients on Spanish STSB across all layers.

C Performance for Various Model Sizes on STSB

We further strengthen our findings with experiments for various model sizes. The models used for the experiments are BERT, ELECTRA discriminator and ELECTRA generator. The figures 10, 11, and 12 show the test set Spearman correlation coefficients on STSB across various model sizes. The shaded are in the figures represents the standard deviation.

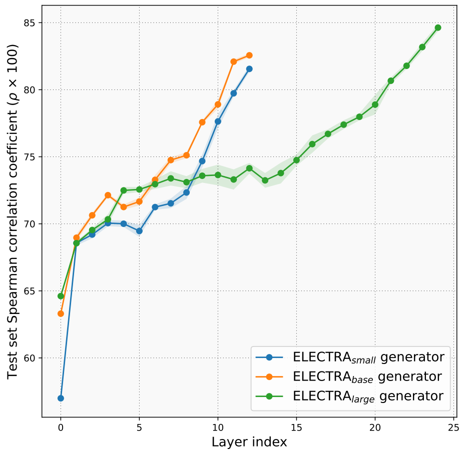


Figure 10: Test set Spearman correlation coefficients on STSB across various ELECTRA generator model sizes.

For the ELECTRA generator experiments (Figure 10), we use the small,[12] base,[13] and large[14] sizes. The highest Spearman correlation coefficient scores are 81.55, 82.57, and 84.63, respectively. All highest scores are achieved for the final hidden state, which is in line with other experiments.

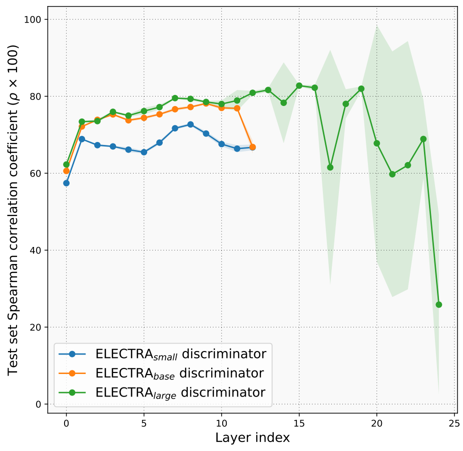


Figure 11: Test set Spearman correlation coefficients on STSB across various ELECTRA discriminator model sizes.

The ELECTRA discriminator experiments (Figure 11) show a performance drop present in the later layers, in agreement with previous results. The highest achieved Spearman correlation coefficients for the small,[15] base,[16] and large[17] sizes are 72.68, 78.14, and 82.77, achieved in layers 8, 9, and 15, respectively.

For experiments with BERT (Figure 12), there are more available standard sizes: tiny, mini, small, medium, base, and large. We apply our method to all of these sizes. The highest results are achieved for the last hidden state, while the scores for tiny, mini, small, medium, base, and large models are 69.80, 75.55, 79.13, 80.74, 82.74, and 85.47, respectively.

[12]https://huggingface.co/google/electra-small-generator

[13]https://huggingface.co/google/electra-base-generator

[14]https://huggingface.co/google/electra-large-generator

[15]https://huggingface.co/google/electra-small-discriminator

[16]https://huggingface.co/google/electra-base-discriminator

[17]https://huggingface.co/google/electra-large-discriminator

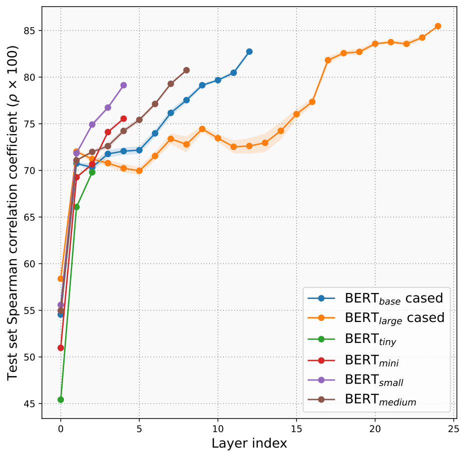


Figure 12: Test set Spearman correlation coefficients on STSB across various BERT model sizes.

D Overview of Best-Performing Models Using TMFT on STSB

Table 2 represents the test set Spearman correlation coefficients and Pearson correlation coefficients on STSB for models that achieve the highest validation Spearman correlation coefficient using TMFT. The Table also includes the number of parameters of the truncated model. Bold values represent the highest values for the model family. BERT achieves the highest Spearman correlation coefficient for the last hidden layer, which is in line with other experiments. BERT$_{\text{large}}$ achieves the highest test set Spearman correlation coefficient of 85.47 with 332.53M parameters. For the ELECTRA discriminator small, base, and large, the hidden states with the highest validation Spearman correlation coefficient are 1, 3, and 12, respectively. The ELECTRA discriminator greatly benefits from model truncation, which is demonstrated by the improvements of the Spearman correlation coefficient by 2.16, 8.47, and 55.06 points, while using 8.69M, 63.79M, and 151.15M parameters less for the small, base, and large models, respectively. Finally, the ELECTRA generator model family provides a parameter efficient alternative to BERT. The largest ELECTRA generator model uses only 50.74M parameters and achieves a test set Spearman correlation coefficient of 84.63, only 0.84 points less than BERT$_{\text{large}}$, which has 281.79M more parameters.

Model	Layer	Params	Val	Test
$\text{BERT}_{\text{tiny}}$	2	4.37M	78.20/77.57	69.80/70.64
$\text{BERT}_{\text{mini}}$	4	11.10M	83.06/82.42	75.55/76.28
$\text{BERT}_{\text{small}}$	4	28.50M	85.25/85.09	79.13/79.56
$\text{BERT}_{\text{medium}}$	8	41.11M	85.74/85.46	80.74/81.02
$\text{BERT}_{\text{base}}$	12	107.72M	86.07/85.98	82.74/83.03
$\text{BERT}_{\text{large}}$	24	332.53M	**88.33/88.31**	**85.47/85.68**
$\text{ELECTRA}_{\text{D small}}$	1	4.76M	79.74/79.27	68.88/69.64
$\text{ELECTRA}_{\text{D small last}}$	12	13.45M	73.98/73.14	66.72/67.27
$\text{ELECTRA}_{\text{D base}}$	3	45.10M	82.15/82.20	75.29/76.96
$\text{ELECTRA}_{\text{D base last}}$	12	108.89M	72.41/71.62	66.82/67.23
$\text{ELECTRA}_{\text{D large}}$	12	182.94M	**84.74/84.88**	**80.90/81.15**
$\text{ELECTRA}_{\text{D large last}}$	24	334.09M	29.88/28.44	25.84/25.21
$\text{ELECTRA}_{\text{G small}}$	12	13.45M	84.62/84.11	81.55/80.93
$\text{ELECTRA}_{\text{G base}}$	12	33.31M	86.62/86.38	82.57/82.50
$\text{ELECTRA}_{\text{G large}}$	24	50.74M	**87.23/86.86**	**84.63/84.52**

Table 2: An overview of the test set Spearman correlation coefficients and Pearson correlation coefficients for various model families and model sizes. The reported test set values correspond to the model with the highest validation set Spearman correlation coefficient. For parameter calculation, the pooler layer is excluded as we do not use it.

E TMFT on Randomly Initialized Models

To verify whether excluding the pre-training step will shed more light on the performance drop, we provide an ablation study with randomly initialized models. Our hypothesis is that all models should roughly exhibit the same behaviour when fine-tuned up to a certain layer. Figure 13 presents the result for TMFT on randomly initialized models. After layer zero, all models exhibit the same behaviour, with minor oscillations. The biggest difference is present in layer zero, where ELECTRA generator performs the best. The results suggest that architecture and tokenizer choice are not the cause of the performance drop. However, this does not exclude the effect of pre-training data, pre-training method, fine-tuning data, or hyperparameter choice.

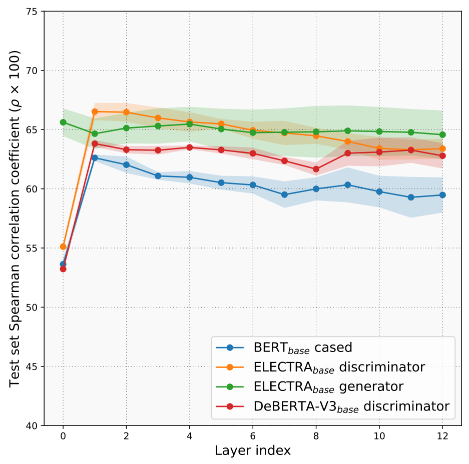


Figure 13: Test set Spearman correlation coefficients on STSB for randomly initialized models.

F Reproducibility

We conducted our experiments on an AMD Ryzen Threadripper 3970X 32-Core Processor and a single RTX 3090 GPU with 24GB of RAM. Running the experiments took around 300 GPU hours. DAPT experiments with BERT and ELECTRA discriminator take around 7 GPU hours each, while for the ELECTRA generator it is around 4.5 GPU hours. Pre-training DeBERTaV3 took the longest, lasting around 30 hours. The word similarity fine-tuning takes up to 3 minutes, regardless of the model.