Fine-Tuned Transformers Show Clusters of Similar Representations Across Layers

Anonymous ACL submission

Abstract

Despite the success of fine-tuning pretrained language encoders like BERT for downstream natural language understanding (NLU) tasks, it is still poorly understood how neural networks change after fine-tuning. In this work, we use centered kernel alignment (CKA), a method for comparing learned representations, to measure the similarity of representations in task-tuned models across layers. In experiments across twelve NLU tasks, we discover a consistent block diagonal structure in the similarity of representations within fine-tuned RoBERTa and ALBERT models, with strong similarity within clusters of earlier and later layers, but not between them. The similarity of later layer representations implies that later layers only marginally contribute to task performance, and we verify in experiments that the top few layers of fine-tuned Transformers can be discarded without hurting performance, even with no further tuning.

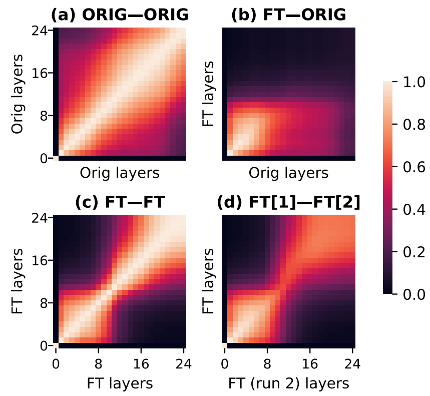

Figure 1: CKA similarity scores of CLS (classifier token) representations of ORIG (untuned ALBERT) and FT (fine-tuned) models on RTE, across different layers of the model. FT[1]–FT[2] compares two RTE models with different random restarts. ORIG–ORIG and FT–FT are symmetric by construction. Fine-tuned models exhibit a block-diagonal structure in the representation similarities. The same color scale is used in all plots.

1 Introduction

Fine-tuning pretrained language encoders such as BERT (Devlin et al., 2019) and its successors (Liu et al., 2019b; Lan et al., 2020; Clark et al., 2020; He et al., 2020) has proven to be highly successful, attaining state-of-the-art performance on many language tasks, but how do these models internally represent task-specific knowledge?

In this work, we study how learned representations change through fine-tuning by studying the similarity of representations between layers of untuned and task-tuned models. We use centered kernel alignment (CKA; Kornblith et al., 2019) to measure representation similarity and conduct extensive experiments across three pretrained encoders and twelve language understanding tasks.

We discover a consistent, block diagonal structure (Figure 1c,d) in the similarity of learned representations for almost all task-tuned RoBERTa and ALBERT models, where early layer representations and later layer representations form two distinct clusters, with high intra-cluster and low inter-cluster similarity.

Given the strong representation similarity of later model layers, we hypothesize that many of the later layers only marginally contribute to task performance. We show in experiments that the later layers of task-tuned RoBERTa and ALBERT can indeed be discarded with minimal impact to performance, even without any further fine-tuning.

2 Experimental Setup

Models For the majority of our experiments, we consider three commonly used language-encoding models: RoBERTa (Liu et al., 2019b), ALBERT (Lan et al., 2020) and ELECTRA (Clark et al., 2020). Because of the large number of experiments being performed, we use RoBERTa$_{\text{BASE}}$, ALBERT$_{\text{LARGEV2}}$ and ELECTRA$_{\text{BASE}}$ rather than the largest available versions of these models.

Tasks We use the tasks included in the GLUE benchmark (Wang et al., 2018) excluding the data-poor WNLI, namely: CoLA (Warstadt et al., 2019), MNLI (Williams et al., 2018), MRPC (Dolan and Brockett, 2005), QNLI (Rajpurkar et al., 2016), QQP,[1] RTE (Dagan et al., 2005), SST-2 (Socher et al., 2013), and STS-B (Cer et al., 2017). We include four additional tasks to cover a more diverse set of task formats and difficulties: BoolQ (Clark et al., 2019) and Yelp Review Polarity (Zhang et al., 2015) classification tasks, and HellaSwag (Zellers et al., 2019) and CosmosQA (Huang et al., 2019) multiple-choice tasks.

Optimization The representations learned over the course of training and similarity of representations may be sensitive to the number of steps used in training. To control for this, and to avoid task-specific hyperparameter tuning, we fine-tune on each task for up to 10,000 steps. We use the Adam (Kingma and Ba, 2014) optimizer with batch size of 4, a learning rate of 1e-5, and 1,000 warmup optimization steps.

We use the `jiant` (Phang et al., 2020) library, built on Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019), to run our experiments.

3 Representation Similarity with CKA

To analyze how learned representations change via fine-tuning, we use centered kernel alignment (CKA; Kornblith et al., 2019) to measure representation similarity. CKA is invariant to both orthogonal transformation and isotropic scaling of the compared representations, making it ideal for measuring the similarity of neural network representations, and has applied to BERT-type models in prior work (Wu et al., 2020; Sridhar and Sarah, 2020). Given two sets of representations $X \in \mathbb{R}^{N \times d_1}$ and $Y \in \mathbb{R}^{N \times d_1}$ where N is the number of examples and d_1, d_2 the hidden dimensions, CKA computes a similarity score between 0 and 1, where a higher score indicates greater similarity. Further details on CKA are provided in Appendix A.

Using CKA, we can compare the similarity of representations between different layers of the same model or even different models. For our analysis, we use the representations of the `CLS` token, i.e. the token whose final layer representation is fed to the task output head.[2] We compute CKA over the validation examples of each task.

To provide intuition for CKA scores, we first show in Figure 1 an example of the comparison formats using ALBERT fine-tuned on RTE.

ORIG–ORIG The top left plot shows the similarity of representations across the layers of the untuned ALBERT model on RTE inputs. Adjacent layers have high similarity scores, only gradually decreasing as more distant layers are compared.

FT–ORIG We show layers of the task-tuned model on the Y-axis and untuned model on the X-axis. The `CLS` representations of the later layers in the task-tuned model appear highly dissimilar to any of the untuned model: In other words, the representations differ starkly from those used for ALBERT's masked language modeling (MLM) and sentence order prediction (SOP) pretraining. This coheres with prior work showing that representations of later layers are most likely to change during fine-tuning (Kovaleva et al., 2019; Wu et al., 2020).

FT–FT Next, we compare layers within a single fine-tuned model. We observe a block-diagonal structure in the representation similarities—two distinct clusters of earlier (approx. first 10) and later (approx. last 14) layers that have high inter-cluster but low intra-cluster similarity. When considered together with FT-ORIG, we can infer that the earlier layer representations resemble those used for pretraining, whereas the later layers encode a representation suitable for tackling the task. The high internal similarity between the top few layers and the sharp block diagonal structure of the similarity matrix imply that the representations starkly differ.

FT[1]–FT[2] Finally, we compare fine-tuned ALBERT models across two random restarts. We observe a similar block diagonal structure. In particular, the similarity of the `CLS` representations in the later layers indicates that CKA is able recover the similarity of representations for tackling the same task across random restarts.

3.1 Results

We extend our CKA analysis to all twelve tasks and all three pretrained models, showing the FT-FT results in Figure 2. We observe that the block diagonal structure of representation similarity identified

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[2] RoBERTa uses a <s> token instead, but for brevity and consistency, we will refer to it as `CLS` as well.

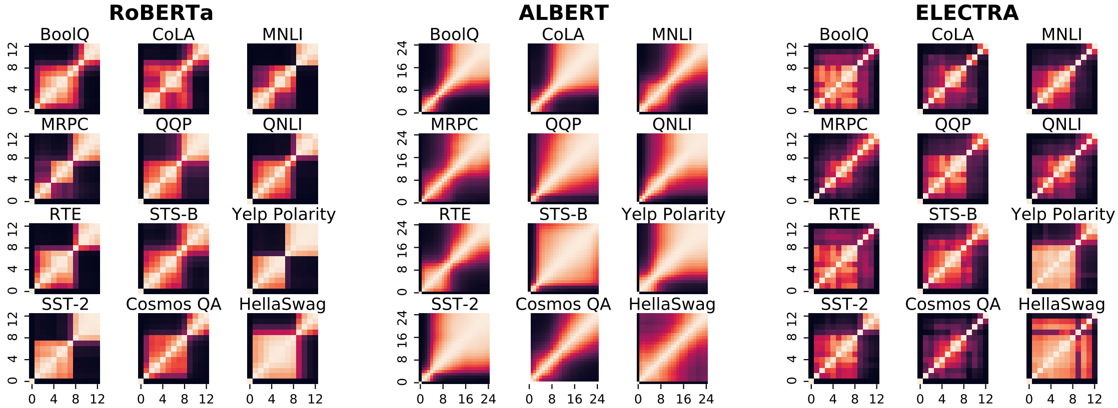

Figure 2: Representation similarity between layers for task-tuned models (FT–FT). RoBERTa and ALBERT task models exhibit a 'block diagonal' structure in the representation similarity of `CLS` tokens across nearly all tasks.

in Section 3 appears in almost every RoBERTa and ALBERT model, sharply delineating the earlier and later clusters of representations. In fact, RoBERTa often has even more distinct clusters than ALBERT. We hypothesize that since ALBERT shares parameters across layers, it is more difficult for representations to sharply change across a single layer, whereas RoBERTa, which has no parameter sharing, has no such constraint.

The significant similarity of the later layers suggests that many of the later layers may not contribute much to the task. Given residual connections between Transformer layers, later layers could learn a 'no-op' or only slightly adjust the output representation if the task can be adequately 'solved' at an earlier layer. If this is true, we should be able to feed an intermediate representation from later layers to the output head with no further fine-tuning and retain most of the task performance. We investigate this hypothesis in Section 4.

In contrast, we do not see the same pattern in the ELECTRA models. The representations of the later layers are generally highly dissimilar even up to the penultimate layer in many tasks. A few tasks do exhibit a minor block diagonal structure, such as STS-B, Yelp Polarity and SST-2, but it is far less apparent compared to the other two models. ELECTRA has a very different pretraining task from the other two models (replaced token detection), which may explain this difference.

We see complementary results for FT–ORIG and FT[1]–FT[2] in Figure 4 and Figure 5. For RoBERTa and ALBERT, while the earlier layers of the task models have similar `CLS` representations to the untuned models, the later layers are largely dissimilar to any layer in the base model.

4 Truncating Fine-tuned Models

To test our hypothesis that the later layers of tuned task-models only marginally contribute to task performance, we propose a simple experiment where we feed the representations from an intermediate layer directly to the task output head, effectively discarding the later layers. We refer to these as *truncated* models. We test three different configurations: (a) UNTUNED , where we feed intermediate representations from a fine-tuned model to the tuned task output head *without any further fine-tuning*, (b) TUNED , where we fine-tune only the output head, and (c) TUNEDORIG , where we use representations from the base model (not fine-tuned on the task), but we fine-tune the output head. Performance of the UNTUNED trunated models indicates the extent to which an intermediate representation can be directly substituted for the final layer's representation; the TUNED and TUNEDORIG models provide an upper-bound of performance using the `CLS` representation of a given layer of a fine-tuned and non-fine-tuned encoder respectively.

Our results are shown in Figure 3. For RoBERTa and ALBERT, we find that the UNTUNED truncated models perform comparably to the Tuned truncated and full fine-tuned models[3] at the later layers. For instance, the top 4 layers of the RoBERTa for Yelp Polarity model can be discarded with no further tuning and minimal impact to performance (95.5 vs 96.1). On the other hand, TUNEDORIG models perform very poorly compared to the TUNED models across all layers, showing that task-tuned intermediate representations are crucial for good per-

[3]An UNTUNED model using the final layer representation is equivalent to a regular fine-tuned model.

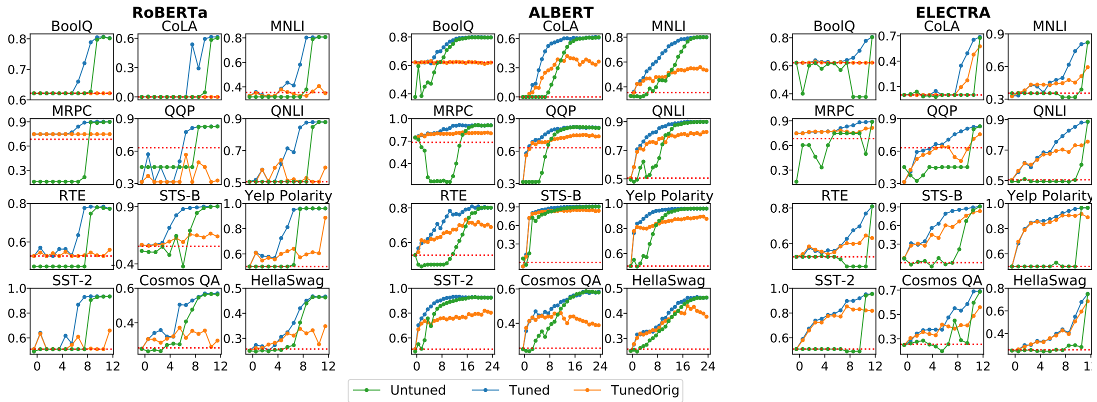

Figure 3: Model Truncation Experiments: Task performance (Y-axis) when feeding representation from an intermediate layer (X-axis) directly to the task output head, equivalent to discarding the top layers of the model. UNTUNED (green), uses a task-tuned encoder, but no further fine-tuning of the task-tuned output head. TUNED (blue), involves further fine-tuning the output head on the intermediate representation. TUNEDORIG (yellow) uses the pretrained encoder, but the output head is fine-tuned. For RoBERTa and ALBERT, the top few layers bcan e discarded for many tasks in either TUNED or UNTUNED configurations without hurting performance. The majority class baseline is shown with a red dotted line.

formance, even when fine-tuning the output head. For ALBERT, which shares parameters between layers, a larger fraction of layers can be discarded with minimal impact to performance for both UNTUNED and TUNED truncated models.

On the other hand, we do not find a similar pattern in ELECTRA models. The UNTUNED truncated models perform extremely poorly when discarding almost any number of layers, and even the TUNED truncated models quickly drop in performance with even one or two layers discarded. These results are consistent with our CKA analyses that showed that the learned and task-tuned representations for ELECTRA do not share the same structure as those of RoBERTa and ALBERT. We speculate that this differences stems from the different pretraining objectives—replaced token detection is a binary prediction problem, whereas masked language modeling involves predicting a distribution over a large number of tokens—leading to differences in learned representations that propagate even to fine-tuned models. We leave further investigation these differences to future work.

5 Related Work

While CKA (Kornblith et al., 2019) was initially proposed as an interpretability method for computer vision models, it has more recently seen application to NLP models. Wu et al. (2020) applied CKA to pretrained Transformers models such as BERT and GPT-2, focusing on cross-model comparison—our analysis builds on their findings, with greater focus on layer-wise comparisons and implications for fine-tuning. Sridhar and Sarah (2020) use CKA to measure the impact of a proposed model architecture change on the learned representations. Voita et al. (2019) and Merchant et al. (2020) apply similar representation similarity analyses to Transformers, with the latter also investigating freezing and dropping layers from models.

More broadly, significant work has been done on better understanding and interpreting the capabilities of BERT-type models—Rogers et al. (2020) offers a thorough survey of this line of work. Of particular relevance to our work: Work on model probing (Tenney et al., 2019b; Liu et al., 2019a; Tenney et al., 2019a) has studied the extent to syntactic and semantic features are represented at different layers of BERT-type models.

6 Conclusion

We show a consistent pattern to the structure of representation similarity in task-tuned RoBERTa and ALBERT models, with strong representation similarity within clusters of earlier and later layers, but not between them. We further show that the later layers of task-tuned RoBERTa and ALBERT models can often be discarded without hurting task performance, verifying that the later layers of these models truly have similar representations. However, we find that ELECTRA models exhibit starkly different properties from the other two models, which prompts further investigation into how and why these models differ.

References

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.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. 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. In *ICLR*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *International Workshop on Paraphrasing*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations, San Diego, 2015*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. Similarity of neural network representations revisited. *CoRR*, abs/1905.00414.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized bert pretraining approach. Unpublished manuscript available on arXiv.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jason Phang, Phil Yeres, Jesse Swanson, Haokun Liu, Ian F. Tenney, Phu Mon Htut, Clara Vania, Alex Wang, and Samuel R. Bowman. 2020. `jiant` 2.0: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Sharath Nittur Sridhar and Anthony Sarah. 2020. Undivided attention: Are intermediate layers necessary for bert?

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. 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 4396–4406, Hong Kong, China. Association for Computational Linguistics.

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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics (TACL)*, 7:625–641.

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. 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.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Centered Kernel Alignment

Given two sets of representations $X \in \mathbb{R}^{N \times d}$ and $Y \in \mathbb{R}^{N \times d}$ where N is the number of examples and d the hidden dimension (for instance the `CLS` vector representations of a set of examples from two different layers of the same model), CKA computes a similarity score between 0 and 1. :

$$\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K)\text{HSIC}(L, L)}}$$

with

$$\text{HSIC}(K, L) = \frac{1}{(n-1)^2}\text{tr}(KHLH)$$

and $H = I_n - \frac{1}{b}\mathbf{1}\mathbf{1}^T$ $K = XX^T$, $L = YY^T$ when using a linear kernel. We refer the reader to the original work (Kornblith et al., 2019) for more details and properties of CKA.

B Additional Results

Figure 4 shows the FT–ORIG plots for all tasks and models.

Figure 5 shows the FT[1]–FT[2] plots for all tasks and models.

Figure 6 computes representation similarity *between* models.

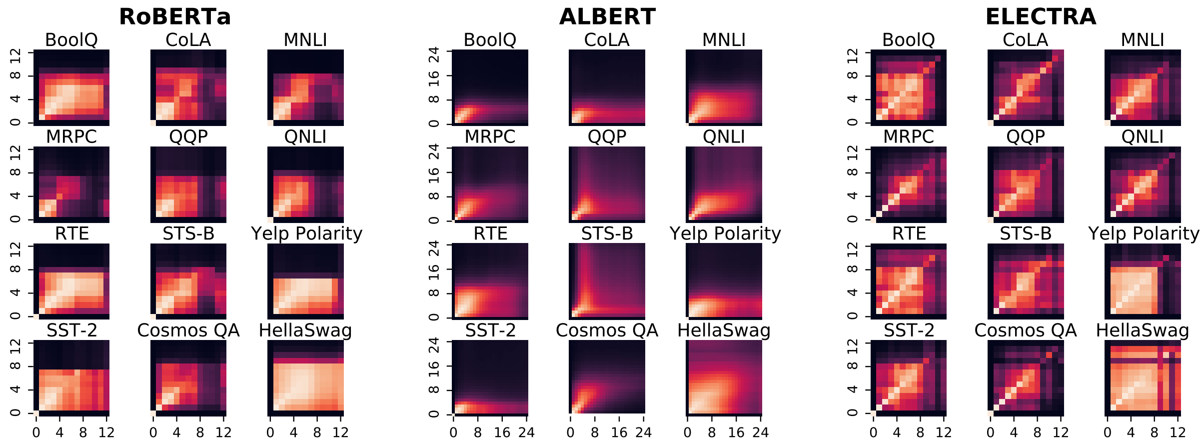

Figure 4: CKA representation similarity for FT–ORIG. Task-tuned layers are on the Y-axis, untuned layers in the X-axis. `CLS` representations of the top few layers RoBERTa and ALBERT models are highly dissimilar to those of the pretrained model at any layer.

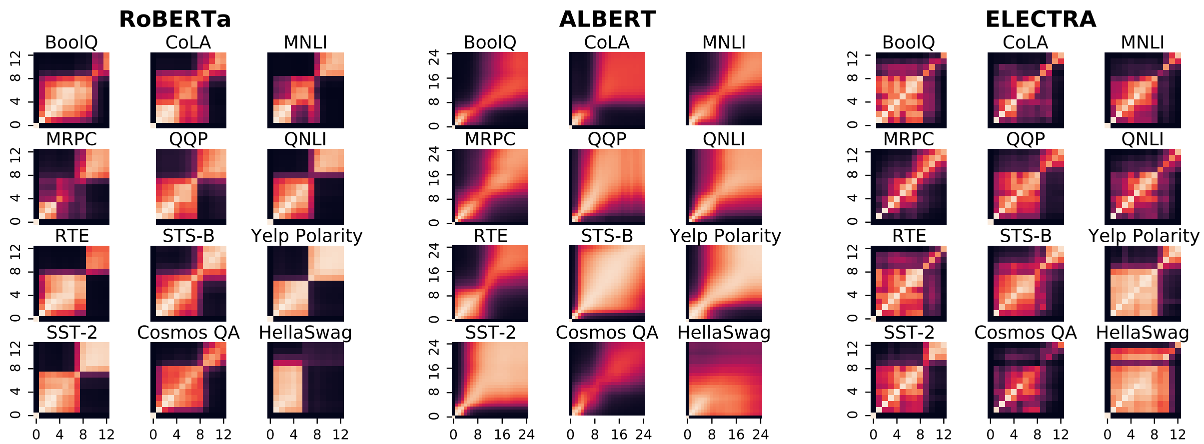

Figure 5: CKA representation similarity for FT[1]–FT[2]. RoBERTa and ALBERT task models exhibit a ‘block diagonal‘ structure to representation similarity of `CLS` tokens, indicating in particular that the representations of the top few layers are highly similar. Plots for tasks that do not use the `CLS` token are dimmed.

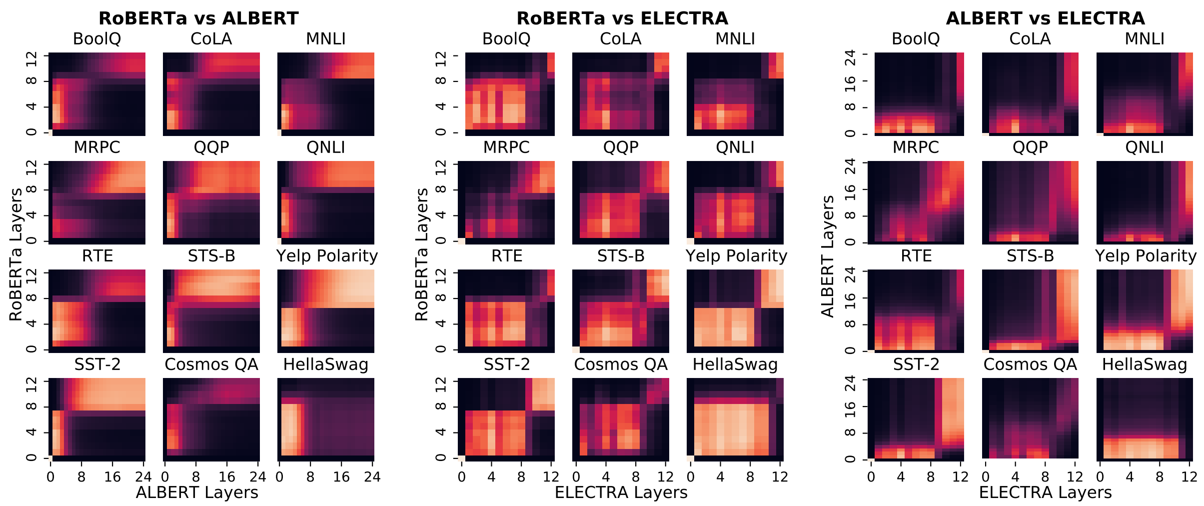

Figure 6: CKA representation similarity comparing `CLS` representations cross models. The upper right blocks indicate the representations in the earlier and the later layers are similar even across models.