# Similarity Analysis of Contextual Word Representation Models

**John M. Wu**[*1] **Yonatan Belinkov**[*12]
**Hassan Sajjad**[3] **Nadir Durrani**[3] **Fahim Dalvi**[3] **James Glass**[1]
[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA
[2]Harvard John A. Paulson School of Engineering and Applied Sciences, Cambridge, MA, USA
[3]Qatar Computing Research Institute, HBKU Research Complex, Doha 5825, Qatar
{johnmwu,belinkov,glass}@csail.mit.edu
{hsajjad,ndurrani,faimaduddin}@qf.org.qa

## Abstract

This paper investigates contextual word representation models from the lens of similarity analysis. Given a collection of trained models, we measure the similarity of their internal representations and attention. Critically, these models come from vastly different architectures. We use existing and novel similarity measures that aim to gauge the level of localization of information in the deep models, and facilitate the investigation of which design factors affect model similarity, without requiring any external linguistic annotation. The analysis reveals that models within the same family are more similar to one another, as may be expected. Surprisingly, different architectures have rather similar representations, but different individual neurons. We also observed differences in information localization in lower and higher layers and found that higher layers are more affected by fine-tuning on downstream tasks.[1]

## 1 Introduction

Contextual word representations such as ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019) have led to impressive improvements in a variety of tasks. With this progress in breaking the state of the art, interest in the community has expanded to analyzing such models in an effort to illuminate their inner workings. A number of studies have analyzed the internal representations in such models and attempted to assess what linguistic properties they capture. A prominent methodology for this is to train supervised classifiers based on the models' learned representations, and predict various linguistic properties. For instance, Liu et al. (2019a) train such classifiers on 16 linguistic tasks, including part-of-speech tagging, chunking, named entity recognition, and others. Such an approach may reveal how well representations from different models, and model layers, capture different properties. This approach, known as analysis by probing classifiers, has been used in numerous other studies (Belinkov and Glass, 2019).

While the above approach yields compelling insights, its applicability is constrained by the availability of linguistic annotations. In addition, comparisons of different models are indirect, via the probing accuracy, making it difficult to comment on the similarities and differences of different models. In this paper, we develop complementary methods for analyzing contextual word representations based on their inter- and intra-similarity. While this similarity analysis does not tell us absolute facts about a model, it allows comparing representations without subscribing to one type of information. We consider several kinds of similarity measures based on different levels of localization/distributivity of information: from neuron-level pairwise comparisons of individual neurons to representation-level comparisons of full word representations. We also explore similarity measures based on models' attention weights, in the case of Transformer models (Vaswani et al., 2017). This approach enables us to ask questions such as: Do different models behave similarly on the same inputs? Which design choices determine whether models behave similarly or differently? Are certain model components more similar than others across architectures? Is the information in a given model more or less localized (encoded in individual components) compared to other models?[2]

[*]Equal contribution

[1]The code is available at https://github.com/johnmwu/contextual-corr-analysis.

[2]Hinton (1984) defines a localist representation as one using one computing element for each represented entity. In a language model, this definition would depend on what linguistic concepts we deem important, and is thus somewhat arbitrary. We develop a measure that aims to capture this notion of localization without recourse to a specific set of linguistic properties.



*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655
July 5 - 10, 2020. 

We choose a collection of pre-trained models that aim to capture diverse aspects of modeling choices, including the building blocks (Recurrent Networks, Transformers), language modeling objective (unidirectional, bidirectional, masked, permutation-based), and model depth (from 3 to 24 layers). More specifically, we experiment with variants of ELMo, BERT, GPT (Radford et al., 2018), GPT2 (Radford et al., 2019), and XLNet (Yang et al., 2019). Notably, we use the same methods to investigate the effect that fine-tuning on downstream tasks has on the model similarities.

Our analysis yields the following insights:

- Different architectures may have similar representations, but different individual neurons. Models within the same family are more similar to one another in terms of both their neurons and full representations.
- Lower layers are more similar than higher layers across architectures.
- Higher layers have more localized representations than lower layers.
- Higher layers are more affected by fine-tuning than lower layers, in terms of their representations and attentions, and thus are less similar to the higher layers of pre-trained models.
- Fine-tuning affects the localization of information, causing high layers to be less localized.

Finally, we show how the similarity analysis can motivate a simple technique for efficient fine-tuning, where freezing the bottom layers of models still maintains comparable performance to fine-tuning the full network, while reducing the fine-tuning time.

## 2 Related Work

The most common approach for analyzing neural network models in general, and contextual word representations in particular, is by probing classifiers (Ettinger et al., 2016; Belinkov et al., 2017; Adi et al., 2017; Conneau et al., 2018; Hupkes et al., 2018), where a classifier is trained on a corpus of linguistic annotations using representations from the model under investigation. For example, Liu et al. (2019a) used this methodology for investigating the representations of contextual word representations on 16 linguistic tasks. One limitation of this approach is that it requires specifying linguistic tasks of interest and obtaining suitable annotations. This potentially limits the applicability of the approach.

An orthogonal analysis method relies on similarities between model representations. Bau et al. (2019) used this approach to analyze the role of individual neurons in neural machine translation. They found that individual neurons are important and interpretable. However, their work was limited to a certain kind of architecture (specifically, a recurrent one). In contrast, we compare models of various architectures and objective functions.

Other work used similarity measures to study learning dynamics in language models by comparing checkpoints of recurrent language models (Morcos et al., 2018), or a language model and a part-of-speech tagger (Saphra and Lopez, 2019). Our work adopts a similar approach, but explores a range of similarity measures over different contextual word representation models.

Questions of localization and distributivity of information have been under investigation for a long time in the connectionist cognitive science literature (Page, 2000; Bowers, 2002; Gayler and Levy, 2011). While neural language representations are thought to be densely distributed, several recent studies have pointed out the importance of individual neurons (Qian et al., 2016; Shi et al., 2016; Radford et al., 2017; Lakretz et al., 2019; Bau et al., 2019; Dalvi et al., 2019; Baan et al., 2019). Our study contributes to this line of work by designing measures of localization and distributivity of information in a collection of models. Such measures may facilitate incorporating neuron interactions in new training objectives (Li et al., 2020).

## 3 Similarity Measures

We present five groups of similarity measures, each capturing a different similarity notion. Consider a collection of $M$ models $\{f^{(m)}\}_{m=1}^{M}$, yielding word representations $\mathbf{h}_l^{(m)}$ and potentially attention weights $\boldsymbol{\alpha}_l^{(m)}$ at each layer $l$. Let $k$ index neurons $\mathbf{h}_l^{(m)}[k]$ or attention heads $\boldsymbol{\alpha}_l^{(m)}[k]$. $\mathbf{h}_l^{(m)}[k]$, $\boldsymbol{\alpha}_l^{(m)}[k]$ are real (resp. matrix) valued, ranging over words (resp. sentences) in a corpus. Our similarity measures are of the form $\texttt{sim}(\mathbf{h}_l^{(m)}, \mathbf{h}_{l'}^{(m')})$ or $\texttt{sim}(\boldsymbol{\alpha}_l^{(m)}, \boldsymbol{\alpha}_{l'}^{(m')})$, that is, they find similarities between layers. We present the full mathematical details in appendix A.

### 3.1 Neuron-level similarity

A neuron-level similarity measure captures similarity between pairs of individual neurons. We consider one such measure, `neuronsim`, following Bau et al. (2019). For every neuron $k$ in layer $l$, `neuronsim` finds the maximum correlation between it and another neuron in another layer $l'$. Then, it averages over neurons in layer $l$.[3] This measure aims to capture localization of information. It is high when two layers have pairs of neurons with similar behavior. This is far more likely when the models have local, rather than distributed representations, because for distributed representations to have similar pairs of neurons the information must be distributed similarly.

### 3.2 Mixed neuron–representation similarity

A mixed neuron–representation similarity measure captures a similarity between a neuron in one model with a layer in another. We consider one such measure, `mixedsim`: for every neuron $k$ in layer $l$, regress to it from all neurons in layer $l'$ and measure the quality of fit. Then, average over neurons in $l$. It is possible that some information is localized in one layer but distributed in another layer. `mixedsim` captures such a phenomenon.

### 3.3 Representation-level similarity

A representation-level measure finds correlations between a full model (or layer) simultaneously. We consider three such measures: two based on canonical correlation analysis (CCA), namely singular vector CCA (`svsim`; Raghu et al. 2017) and projection weighted CCA (`pwsim`; Morcos et al. 2018), in addition to linear centered kernel alignment (`ckasim`; Kornblith et al. 2019).[4] These measures emphasize distributivity of information—if two layers behave similarly over all of their neurons, the similarity will be higher, even if no individual neuron has a similar matching pair or is represented well by all neurons in the other layer.

Other representation-level similarity measures may be useful, such as representation similarity analysis (RSA; Kriegeskorte et al. 2008), which has been used to analyze neural network representations (Bouchacourt and Baroni, 2018; Chrupała and Alishahi, 2019; Chrupała, 2019), or other variants of CCA, such as deep CCA (Andrew et al., 2013). We leave the explorations of such measures to future work.

### 3.4 Attention-level similarity

Previous work analyzing network similarity has mostly focused on representation-based similarities (Morcos et al., 2018; Saphra and Lopez, 2019; Voita et al., 2019a). Here we consider similarity based on attention weights in Transformer models.

Analogous to a neuron-level similarity measure, an attention-level similarity measure finds the most "correlated" other attention head. We consider three methods to correlate heads, based on the norm of two attention matrices $\boldsymbol{\alpha}_l^{(m)}[k]$, $\boldsymbol{\alpha}_{l'}^{(m')}[k']$, their Pearson correlation, and their Jensen–Shannon divergence.[5] We then average over heads $k$ in layer $l$, as before. These measures are similar to `neuronsim` in that they emphasize localization of information—if two layers have pairs of heads that are very similar in their behavior, the similarity will be higher.

### 3.5 Distributed attention-level similarity

We consider parallels of the representation-level similarity. To compare the entire attention heads in two layers, we concatenate all weights from all heads in one layer to get an attention representation. That is, we obtain attention representations $\boldsymbol{\alpha}_l^{(m)}[h]$, a random variable ranging over pairs of words in the same sentence, such that $\boldsymbol{\alpha}_{l,(i,j)}^{(m)}[h]$ is a scalar value. It is a matrix where the first axis is indexed by word pairs, and the second by heads. We flatten these matrices and use `svsim`, `pwsim`, and `ckasim` as above for comparing these attention representations. These measures should be high when the entire set of heads in one layer is similar to the set of heads in another layer.

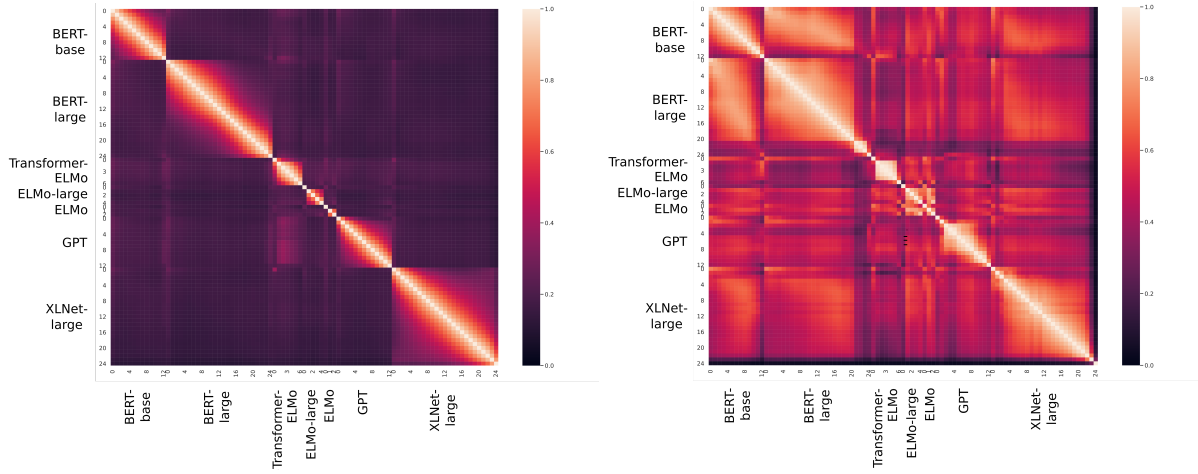
## 4 Experimental Setup

**Models** We choose a collection of pre-trained models that aim to capture diverse aspects of modeling choices, including the building blocks (RNNs, Transformers), language modeling objective (unidirectional, bidirectional, masked, permutation-based), and model depth (from 3 to 24 layers).

[3] In this and other measures that allowed it, we also experimented with averaging just the top $k$ neurons (or canonical correlations, in Section 3.3 measures) in case most of the layer is noise. Heatmaps are in the online repository. We did not notice major differences.

[4] We also experimented with the RBF variant, which is computationally demanding. We found similar patterns in preliminary experiments, so we focus on the linear variant.

[5] Other recent work has used the Jensen–Shannon divergence to measure distances between attention heads (Clark et al., 2019; Jain and Wallace, 2019).




(a) `neuronsim` (b) `ckasim`

Figure 1: Similarity heatmaps of layers in various models under neuron- and representation-level similarities.

*ELMo variants* We use the original ELMo (Peters et al., 2018a), a bidirectional RNN model with two hidden layers, as well as two variants – a deeper and larger 4-layer model and a Transformer-equivalent variant (Peters et al., 2018b).

*GPT variants* We use both the original OpenAI Transformer (GPT; Radford et al. 2018) and its successor GPT2 (Radford et al., 2019), in the small and medium model sizes. These are all unidirectional Transformer LMs.

*BERT* We use BERT-base/large (12/24 layers; Devlin et al. 2019): Transformer LMs trained with a masked LM objective function.[6]

*XLNet* We use XLNet-base/large (12/24 layers; Yang et al. 2019). Both are Transformer LM with a permutation-based objective function.

**Data** For analyzing the models, we run them on the Penn Treebank development set (Marcus et al., 1993), following the setup taken by Liu et al. (2019a) in their probing classifier experiments.[7] We collect representations and attention weights from each layer in each model for computing the similarity measures. We obtain representations for models used in Liu et al. (2019a) from their implementation and use the transformers library (Wolf et al., 2019) to extract other representations. We aggregate sub-word representations by taking the representation of the last sub-word, following Liu et al. (2019a), and sub-word attentions by summing up attention to sub-words and averaging attention from sub-words, following Clark et al. (2019), which guarantees that the attention from each word sums to one.

## 5 Similarity of Pre-trained Models

### 5.1 Neuron and representation levels

Figure 1 shows heatmaps of similarities between layers of different models, according to `neuronsim` and `ckasim`. Heatmaps for the other measures are provided in Appendix B. The heatmaps reveal the following insights.

**Different architectures may have similar representations, but different individual neurons** Comparing the heatmaps, the most striking distinction is that `neuronsim` induces a distinctly block-diagonal heatmap, reflecting high intra-model similarities and low inter-model similarities. As `neuronsim` is computed by finding pairs of very similar neurons, this means that within a model, different layers have similar individual neurons, but across models, neurons are very different. In contrast, `ckasim`- show fairly significant similarities across models (high values off the main diagonal), indicating that different models generate similar representations. The most similar cross-model similarities are found by `mixedsim` (Figure 8d in Appendix B), which suggests that individual neurons in one model may be well represented by a linear combination of neurons in another layer. The other representation-level similarities (`ckasim`, `svsim`, and `pwsim`), also show cross-model similarities, albeit to a lesser extent.

[6] BERT is also trained with a next sentence prediction objective, although this may be redundant (Liu et al., 2019b).

[7] As suggested by a reviewer, we verified that the results are consistent when using another dataset (Appendix B.1).

**Models within the same family are more similar** The heatmaps show greater similarity within a model than across models (bright diagonal). Different models sharing the same architecture and objective function, but different depths, also exhibit substantial representation-level similarities – for instance, compare BERT-base and BERT-large or ELMo-original and ELMo-4-layers, under `ckasim` (Figure 1b). The Transformer-ELMo presents an instructive case, as it shares ELMo's bidirectional objective function but with Transformers rather than RNNs. Its layers are mostly similar to themselves and the other ELMo models, but also to GPT, more so than to BERT or XLNet, which use masked and permutation language modeling objectives, respectively. Thus it seems that the objective has a considerable impact on representation similarity.[8]

The fact that models within the same family are more similar to each other supports the choice of Saphra and Lopez (2019) to use models of similar architecture when probing models via similarity measures across tasks.[9] A possible confounder is that models within the same family are trained on the same data, but cross-family models are trained on different data. It is difficult to control for this given the computational demands of training such models and the current practice in the community of training models on ever increasing sizes of data, rather than a standard fixed dataset. However, Figure 2 shows similarity heatmaps of layers from pre-trained and randomly initialized models using `ckasim`, exhibiting high intra-model similarities, as before. Interestingly, models within the same family (either GPT2 or XLNet) are more similar than across families, even with random models, indicating that intrinsic aspects of models in a given family make them similar, regardless of the training data or process.[10] As may be expected, in most cases, the similarity between random and pre-trained models is small. One exception is the vertical bands in the lower triangle, which indicate that the bottom layers of trained models are similar to many layers of random models. This may be due to random models merely transferring information from bottom to top, without meaningful processing.

[8] Voita et al. (2019a) found that differences in the training objective result in more different representations (according to `pwsim`) than differences in random initialization.

[9] We thank a reviewer for pointing out this connection.

[10] Relatedly, Morcos et al. (2018) found similar CCA coefficients in representations from recurrent language models trained on different datasets.

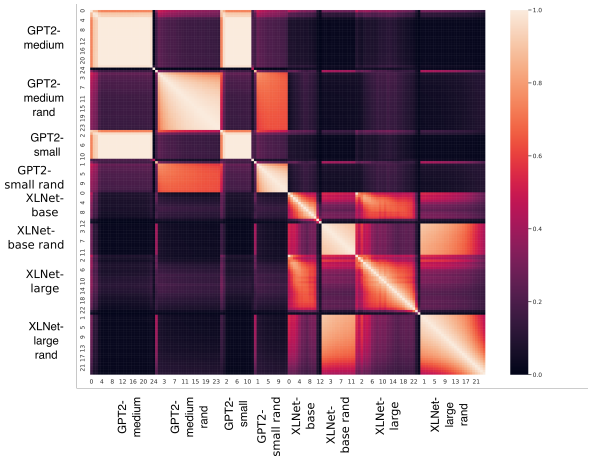



Figure 2: `ckasim` similarity heatmap of layers in base and random models.

Still, it may explain why random models sometimes generate useful features (Wieting and Kiela, 2019). Meanwhile, as pointed out by a reviewer, lower layers converge faster, leaving them closer to their initial random state (Raghu et al., 2017; Shwartz-Ziv and Tishby, 2017).

**Lower layers are more similar across architectures** The representation-level heatmaps (Figure 1) all exhibit horizontal stripes at lower layers, especially with `ckasim`, indicating that lower layers are more similar than higher layers when comparing across models. This pattern can be explained by lower layers being closer to the input, which is always the same words. A similar observation has been made for vision networks (Raghu et al., 2017).[11] Voita et al. (2019a) found a similar pattern comparing Transformer models with different objective functions.

**Adjacent layers are more similar** All heatmaps in Figure 1 exhibit a very bright diagonal and bright lines slightly off the main diagonal, indicating that adjacent layers are more similar. This is even true when comparing layers of different models (notice the diagonal nature of BERT-base vs. BERT-large in Figure 1b), indicating that layers at the same relative depth are more similar than layers at different relative depths. A similar pattern was found in vision networks (Kornblith et al., 2019). Some patterns are unexpected. For instance, comparing

[11] Raghu et al. (2017) also used `svsim` to study recurrent language models, showing that lower layers converge faster. Although they have not looked at cross-model comparisons, faster convergence may be consistent with fewer changes during training, which can explain why lower layers are more similar across architectures.

XLNet with the BERT models, it appears that lower layers of XLNet are more similar to higher layers of BERT. We speculate that this is an artifact of the permutation-based objective in XLNet.

We found corroborating evidence for this observation in ongoing parallel work, where we compare BERT and XLNet at different layers through word- (Liu et al., 2019a) and sentence-level tasks (Wang et al., 2019): while BERT requires mostly features from higher layers to achieve state-of-the-art results, in XLNet lower and middle layers suffice.

**Higher layers are more localized than lower ones** The different similarity measures capture different levels of localization vs. distributivity of information. `neuronsim` captures cases of localized information, where pairs of neurons in different layers behave similarly. `svsim` captures cases of distributed information, where the full layer representation is similar. To quantify these differences, we compute the average similarity according to each measure when comparing each layer to all other layers. In effect, we take the column-wise mean of each heatmap. We do this separately for `svsim` as the distributed measure and `neuronsim` as the localized measure, and we subtract the `svsim` means from the `neuronsim` means. This results in a measure of localization per layer. Figure 3 shows the results.

In all models, the localization score mostly increases with layers, indicating that information tends to become more localized at higher layers.[12] This pattern is quite consistent, but may be surprising given prior observations on lower layers capturing phenomena that operate at a local context (Tenney et al., 2019), which presumably require fewer neurons. However, this pattern is in line with observations made by Ethayarajh (2019), who reported that upper layers of pre-trained models produce more context-specific representations. There appears to be a correspondence between our localization score and Ethayarajh's context-specificity score, which is based on the cosine similarity of representations of the same word in different contexts. Thus, more localized representations are also more context-specific. A direct comparison between context-specificity and localization may be fruitful avenue for future work.

Some models seem less localized than others, especially the ELMo variants, although this may be confounded by their being shallower models. BERT and XLNet models first decrease in localization and then increase. Interestingly, XLNet's localization score decreases towards the end, suggesting that its top layer representations are less context-specific.

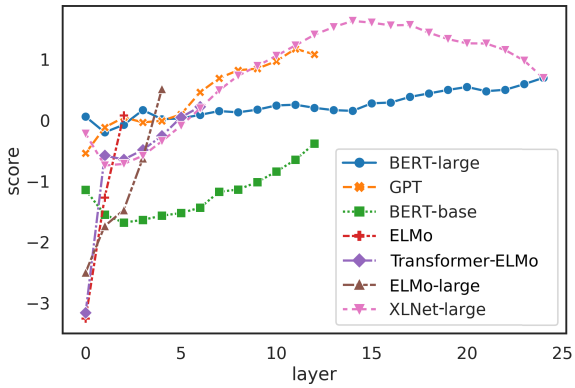


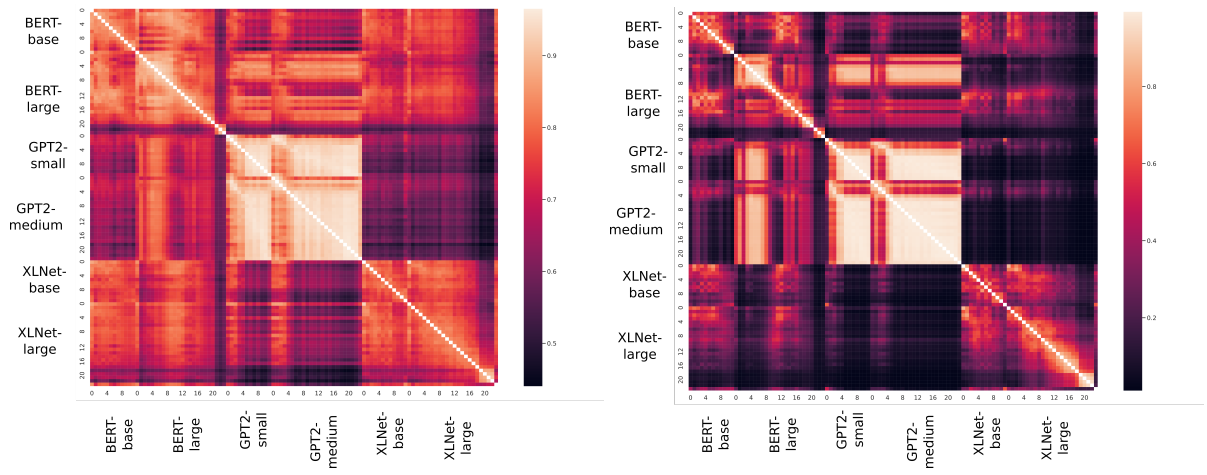

Figure 3: Localization score of various model layers.

## 5.2 Attention level

Figure 4 shows similarity heatmaps using two of the attention-level similarity measures—Jensen–Shannon and `ckasim`—for layers from 6 models: BERT-base/large, GPT2-small/medium, and XLNet-base/large. Layers within the same model or model family exhibit higher similarities (bright block diagonal), in line with results from the representation-level analysis. In particular, under both measures, GPT2 layers are all very similar to each other, except for the bottom ones. Comparing the two heatmaps, the localized Jensen–Shannon similarity (Figure 4a) shows higher similarities off the main diagonal than the distributed `ckasim` measure (Figure 4b), indicating that different models have pairs of attention heads that behave similarly, although the collection of heads from two different models is different in the aggregate. Heatmaps for the other measures are provided in Appendix C, following primarily the same patterns.

It is difficult to identify patterns within a given model family. However, under the attention-based `svsim` (Figure 10d in Appendix C), and to a lesser extent `pwsim` (Figure 10e), we see bright diagonals when comparing different GPT2 (and to a lesser extent XLNet and BERT) models, such that layers at the same relative depth are similar in their attention patterns. We have seen such a result also in the representation-based similarities.

[12] Recurrent models are more monotonous than Transformers, echoing results by Liu et al. (2019a) on language modeling perplexity in different layers.




(a) Jensen–Shannon (b) `ckasim`

Figure 4: Similarity heatmaps of layers in various models under two attention-level similarity measures.

Adjacent layers seem more similar in some cases, but these patterns are often swamped by the large intra-model similarity. This result differs from our results for representational similarity.

GPT2 models, at all layers, are similar to the bottom layers of BERT-large, expressed in bright vertical bands. In contrast, GPT2 models do not seem to be especially similar to XLNet. Comparing XLNet and BERT, we find that lower layers of XLNet are quite similar to higher layers of BERT-base and middle layers of BERT-large. This parallels the findings from comparing representations of XLNet and BERT, which we conjecture is the result of the permutation-based objective in XLNet.

In general, we find the attention-based similarities to be mostly in line with the neuron- and representation-level similarities. Nevertheless, they appear to be harder to interpret, as fine-grained patterns are less noticeable. One might mention in this context concerns regarding the reliability of attention weights for interpreting the importance of input words in a model (Jain and Wallace, 2019; Serrano and Smith, 2019; Brunner et al., 2020). However, characterizing the effect of such concerns on our attention-based similarity measures is beyond the current scope.

## 6 Similarity of Fine-tuned Models

How does fine-tuning on downstream tasks affect model similarity? In this section, we compare pre-trained models and their fine-tuned versions. We use four of the GLUE tasks (Wang et al., 2019):

**MNLI** A multi-genre natural language inference dataset (Williams et al., 2018), where the task is to predict whether a premise entails a hypothesis.

**QNLI** A conversion of the Stanford question answering dataset (Rajpurkar et al., 2016), where the task is to determine whether a sentence contains the answer to a question.

**QQP** A collection of question pairs from the Quora website, where the task is to determine whether two questions are semantically equivalent.

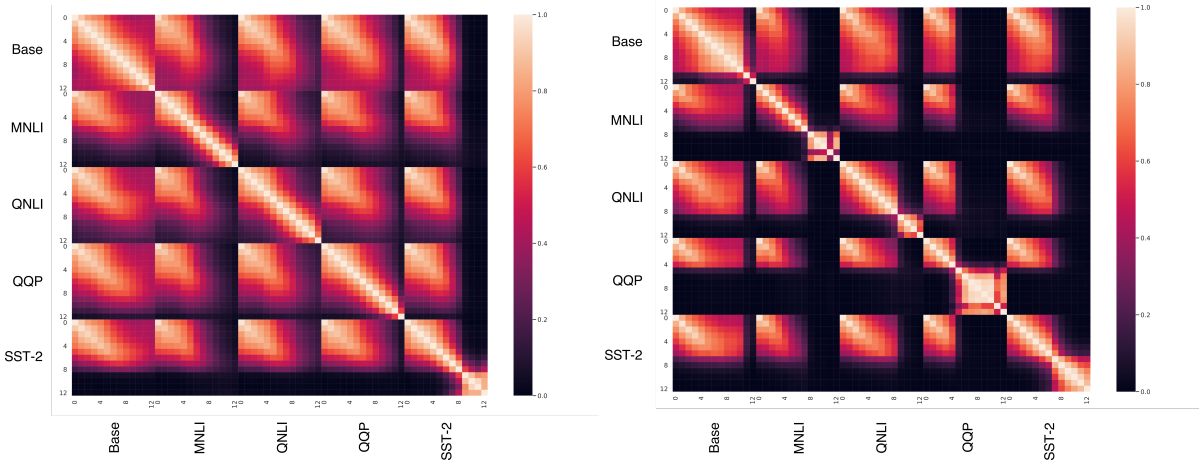
**SST-2** A binary sentiment analysis task using the Stanford sentiment treebank (Socher et al., 2013).

### 6.1 Results

**Top layers are more affected by fine-tuning** Figure 5 shows representation-level `ckasim` similarity heatmaps of pre-trained (not fine-tuned) and fine-tuned versions of BERT and XLNet. The most striking pattern is that the top layers are more affected by fine-tuning than the bottom layers, as evidenced by the low similarity of high layers of the pre-trained models with their fine-tuned counterparts. Hao et al. (2019) also observed that lower layers of BERT are less affected by fine-tuning than top layers, by visualizing the training loss surfaces.[13] In Appendix D, we demonstrate that this observation can motivate a more efficient fine-tuning process, where some of the layers are frozen while others are fine-tuned.

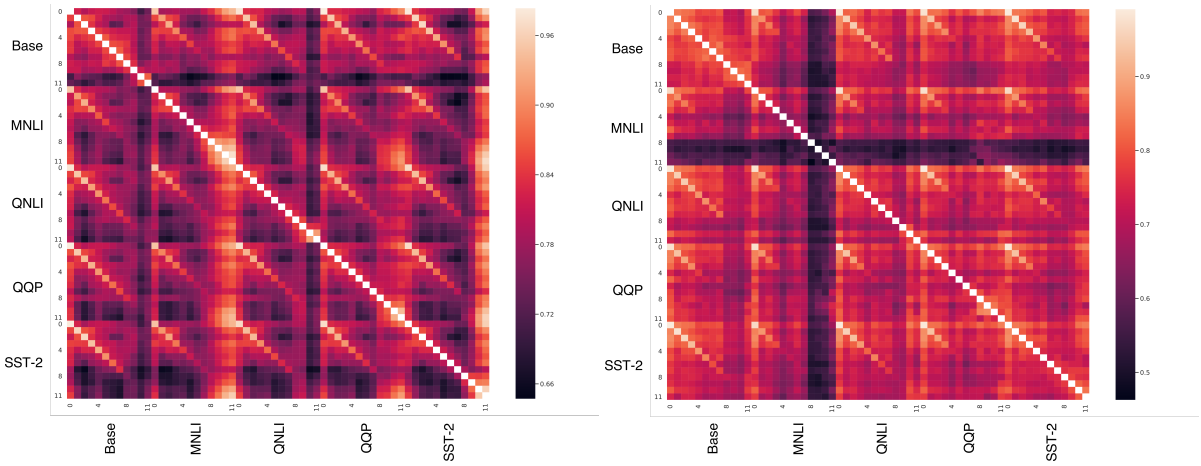
There are some task-specific differences. In BERT, the top layers of the SST-2-fine-tuned model

[13] A reviewer commented that this pattern seems like a natural consequence of back-propagation, which we concur with, although in on-going work we found that middle layers of XLNet lead to more gains when fine-tuned. Future work can also explore the effect of optimization on the similarity measures.




(a) BERT (b) XLNet

Figure 5: `ckasim` similarity heatmaps of layers in base (pre-trained, not fine-tuned) and fine-tuned models.

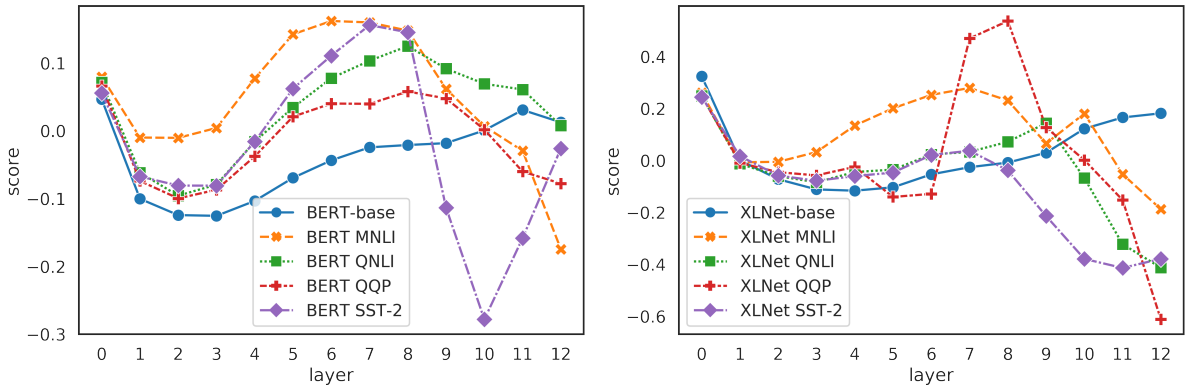



(a) BERT (b) XLNet

Figure 6: Jensen–Shannon attention similarity heatmaps of layers in base (pre-trained, not fine-tuned) and fine-tuned models.

are affected more than other layers. This may be because SST-2 is a sentence classification task, while the other tasks are sentence-pair classification. A potential implication of this is that non-SST-2 tasks can contribute to one another in a multi-task fine-tuning setup. In contrast, in XLNet, fine-tuning on any task leads to top layers being very different from all layers of models fine-tuned on other tasks. This suggests that XLNet representations become very task-specific, and thus multi-task fine-tuning may be less effective with XLNet than with BERT.

Observing the `attnsim` similarity based on Jensen–Shannon divergence for base and fine-tuned models (Figure 6), we again see that top layers have lower similarities, implying that they undergo greater changed during fine-tuning. Other attention-based measures behaved similarly (not shown). Kovaleva et al. (2019) made a similar observation by comparing the cosine similarity of attention matrices in BERT, although they did not perform cross-task comparisons. In fact, the diagonals within each block indicate that bottom layers remain similar to one another even when fine-tuning on different tasks, while top layers diverge after fine-tuning. The vertical bands at layers 0 mean that many higher layers have a head that is very similar to a head from the first layer, that is, a form of redundancy, which can explain why many heads can be pruned (Michel et al., 2019; Voita et al., 2019b; Kovaleva et al., 2019). Comparing BERT and XLNet, the vertical bands at the top layers of BERT (especially in MNLI, QQI, and SST-2) suggest that some top layers are very similar to any other layer. In XLNet, top MNLI layers are quite




(a) BERT (b) XLNet

Figure 7: Localization scores per layer in base and fine-tuned models.

different from any other layer. Thus different objective functions impact the attention heads differently under fine-tuning.

**Fine-tuning affects localization** Figure 7 shows localization scores for different layers in pre-trained and fine-tuned models. In contrast to the pre-trained models, the fine-tuned ones decrease in localization at the top layers. This decrease may be the result of top layers learning high-level tasks, which require multiple neurons to capture properly.

## 7 Conclusion

In this work, we analyzed various prominent contextual word representations from the perspective of similarity analysis. We compared different layers of pre-trained models using both localized and distributed measures of similarity, at neuron, representation, and attention levels. We found that different architectures often have similar internal representations, but differ at the level of individual neurons. We also observed that higher layers are more localized than lower ones. Comparing fine-tuned and pre-trained models, we found that higher layers are more affected by fine-tuning in their representations and attention weights, and become less localized. These findings motivated experimenting with layer-selective fine-tuning, where we were able to obtain good performance while freezing the lower layers and only fine-tuning the top ones.

Our approach is complementary to the linguistic analysis of models via probing classifiers. An exciting direction for future work is to combine the two approaches in order to identify which linguistic properties are captured in model components that are similar to one another, or explicate how localization of information contributes to the learnability of particular properties. It may be insightful to compare the results of our analysis to the loss surfaces of the same models, especially before and after fine-tuning (Hao et al., 2019). One could also study whether a high similarity entail that two models converged to a similar solution. Our localization score can also be compared to other aspects of neural representations, such as gradient distributions and their relation to memorization/generalization (Arpit et al., 2017). Finally, the similarity analysis may also help improve model efficiency, for instance by pointing to components that do not change much during fine-tuning and can thus be pruned.

## Acknowledgements

We thank Nelson Liu for providing some of the representations analyzed in this work. We also thank the anonymous reviewers for their many valuable comments. This research was carried out in collaboration between the HBKU Qatar Computing Research Institute (QCRI) and the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). Y.B. is also supported by the Harvard Mind, Brain, and Behavior Initiative (MBB).

## References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of the Internaltional Confernece for Learning Representations (ICLR)*.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of*

*Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA. PMLR.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242, International Convention Centre, Sydney, Australia. PMLR.

Joris Baan, Jana Leible, Mitja Nikolaus, David Rau, Dennis Ulmer, Tim Baumgärtner, Dieuwke Hupkes, and Elia Bruni. 2019. On the realization of compositionality in neural networks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 127–137, Florence, Italy. Association for Computational Linguistics.

D. Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations (ICLR)*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 981–985, Brussels, Belgium. Association for Computational Linguistics.

Jeffrey S Bowers. 2002. Challenging the widespread assumption that connectionism and distributed representations go hand-in-hand. *Cognitive Psychology*, 45(3):413 – 445.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *International Conference on Learning Representations*.

Grzegorz Chrupała. 2019. Symbolic inductive bias for visually grounded learning of spoken language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6462, Florence, Italy. Association for Computational Linguistics.

Grzegorz Chrupała and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep NLP models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

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)*, 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.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages

1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Ross W. Gayler and Simon D. Levy. 2011. Compositional connectionism in cognitive science ii: the localist/distributed dimension. *Connection Science*, 23(2):85–89.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness 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 4141–4150, Hong Kong, China. Association for Computational Linguistics.

Geoffrey E Hinton. 1984. Distributed representations. Technical Report CMU-CS-84-157, Carnegie Mellon University.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529, Long Beach, California, USA. PMLR.

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 4364–4373, Hong Kong, China. Association for Computational Linguistics.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Jian Li, Xing Wang, Baosong Yang, Shuming Shi, Michael R Lyu, and Zhaopeng Tu. 2020. Neuron interaction based representation composition for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

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. *arXiv preprint arXiv:1907.11692*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? 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 14014–14024. Curran Associates, Inc.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5727–5736. Curran Associates, Inc.

Mike Page. 2000. Connectionist modelling in psychology: A localist manifesto. *Behavioral and Brain Sciences*, 23(4):443âĂŞ467.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. 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.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Analyzing linguistic knowledge in sequential model of sentence. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 826–835, Austin, Texas. Association for Computational Linguistics.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6078–6087. Curran Associates, Inc.

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.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Xing Shi, Kevin Knight, and Deniz Yuret. 2016. Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282, Austin, Texas. Association for Computational Linguistics.

Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

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 EMNLP*, pages 1631–1642.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. 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 4395–4405, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019b. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. In *International Conference on Learning Representations*.

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, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. 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 5753–5763. Curran Associates, Inc.

## A Mathematical Details of Similarity Measures

We assume a fixed corpus with $W = \sum_i W_i$ total words, and $W^{(2)} = \sum_i W_i^2$ total pairs. Here $W_i$ is the number of words in sentence $i$.

A representational layer $\mathbf{h}_l^{(m)}$ may be seen as a $W \times N_m$ matrix, where $N_m$ is the number of neurons (per layer) in model $m$. A single neuron $\mathbf{h}_l^{(m)}[k]$ (really $\mathbf{h}_l^{(m)}[:, k]$) is a $W \times 1$ column vector.

An attention head $\boldsymbol{\alpha}_l^{(m)}[k]$ may be seen as a random variable ranging over sentences $s_i$ and taking matrix values $\boldsymbol{\alpha}_l^{(m)}[k](s_i) \in \mathbb{R}^{t_i \times t_i}$, $t_i = \text{len}(s_i)$.

### A.1 Neuron-level similarity

For a given neuron $\mathbf{h}_l^{(m)}[k]$, we define

$$\texttt{neuronsim}(\mathbf{h}_l^{(m)}[k], \mathbf{h}_{l'}^{(m')}) = \max_{k'} |\rho(\mathbf{h}_{l'}^{(m')}[k'], \mathbf{h}_l^{(m)}[k])|$$

as the maximum correlation between it and another neuron in some layer (Bau et al., 2019). Here $\rho$ is the Pearson correlation. This naturally gives rise to an aggregate measure at the layer level:

$$\texttt{neuronsim}(\mathbf{h}_l^{(m)}, \mathbf{h}_{l'}^{(m')}) = \frac{1}{N_m} \sum_k \texttt{neuronsim}(\mathbf{h}_l^{(m)}[k], \mathbf{h}_{l'}^{(m')})$$

### A.2 Mixed neuron–representation similarity

We define

$$\texttt{mixedsim}(\mathbf{h}_l^{(m)}[k], \mathbf{h}_{l'}^{(m')}) := \text{lstsq}(\mathbf{h}_{l'}^{(m')}, \mathbf{h}_l^{(m)}[k])\texttt{.r}$$

where `.r` is the r-value associated with the regression, the norm of the prediction divided by the norm of the regressand. As before, this is extended to the layer level:

$$\texttt{mixedsim}(\mathbf{h}_l^{(m)}, \mathbf{h}_{l'}^{(m')}) = \frac{1}{N_m} \sum_k \texttt{mixedsim}(\mathbf{h}_l^{(m)}[k], \mathbf{h}_{l'}^{(m')})$$

### A.3 Representation-level similarity

In the following, let $\mathbf{Z}$ denote a column centering transformation. For a given matrix $\mathbf{A}$, the sum of each column in $\mathbf{ZA}$ is zero.

**SVCCA** Given two layers

$$\mathbf{X}, \mathbf{Y} = \mathbf{Z}\mathbf{h}_{l_x}^{(m_x)}, \mathbf{Z}\mathbf{h}_{l_y}^{(m_y)}$$

we compute the truncated principal components

$$\mathbf{X}', \mathbf{Y}' = \mathbf{U_x}[:, : l_x], \mathbf{U_y}[:, : l_y]$$

where $\mathbf{U_x}$ are the left singular vectors of $\mathbf{X}$, and $l_x$ is the index required to account for 99% of the variance. $\mathbf{U_y}$ and $l_y$ are defined analogously. The SVCCA correlations, $\rho_{SVCCA}$, are defined as:

$$\mathbf{u}, \rho_{SVCCA}, \mathbf{v} = \text{SVD}(\mathbf{X'}^T \mathbf{Y'})$$

The SVCCA similarity, $\texttt{svsim}(\mathbf{h}_{l_x}^{(m_x)}, \mathbf{h}_{l_y}^{(m_y)})$, is the mean of $\rho_{SVCCA}$.

**PWCCA** Identical to SVCCA, except the computation of similarity is a weighted mean. Using the same notation as above, we define canonical vectors,

$$\mathbf{H_X} := \mathbf{X'u}$$
$$\mathbf{H_Y} := \mathbf{Y'v}$$

We define alignments

$$\mathbf{A_X} := \text{abs}\left(\mathbf{H_X^T X}\right)$$
$$\mathbf{A_Y} := \text{abs}\left(\mathbf{H_Y^T Y}\right)$$

where abs is the element-wise absolute value. The weights are

$$\alpha_x := \text{weights}(\mathbf{A_X 1}), \quad \alpha_y := \text{weights}(\mathbf{A_Y 1})$$

where $\mathbf{1}$ is the column vector of all ones, and weights normalizes a vector to sum to 1. The PWCCA similarity is

$$\texttt{pwsim}(\mathbf{h}_{l_x}^{(m_x)}, \mathbf{h}_{l_y}^{(m_y)}) := \alpha_x^T \rho_{SVCCA}$$
$$\texttt{pwsim}(\mathbf{h}_{l_y}^{(m_y)}, \mathbf{h}_{l_x}^{(m_x)}) := \alpha_y^T \rho_{SVCCA}$$

It is asymmetric.

**CKA** We use the same notation as above. Given two layers,

$$\mathbf{X}, \mathbf{Y} = \mathbf{Z}\mathbf{h}_{l_x}^{(m_x)}, \mathbf{Z}\mathbf{h}_{l_y}^{(m_y)}$$

the CKA similarity is

$$\texttt{ckasim}(\mathbf{h}_{l_x}^{(m_x)}, \mathbf{h}_{l_y}^{(m_y)}) := \frac{\|\mathbf{X}^T \mathbf{Y}\|^2}{\|\mathbf{X}^T \mathbf{X}\| \, \|\mathbf{Y}^T \mathbf{Y}\|}$$

where $\|\cdot\|$ is the Frobenius norm. It is symmetric.

### A.4 Attention-level similarity

We define

$$\texttt{attnsim}(\boldsymbol{\alpha}_l^{(m)}[k], \boldsymbol{\alpha}_{l'}^{(m')}) = \max_{k'} \left[\text{Sim}(\boldsymbol{\alpha}_{l'}^{(m')}[k'], \boldsymbol{\alpha}_l^{(m)}[k])\right]$$

We consider three such values of Sim.

- Matrix norm: for each sentence $s_i$, compute the Frobenius norm $\left\|\boldsymbol{\alpha}_{l'}^{(m')}[h'](s_i) - \boldsymbol{\alpha}_l^{(m)}[h](s_i)\right\|$. Then average over sentences in the corpus.
- Pearson correlation: for every word $x_i$, compare the attention distributions the two heads

induce from $x_i$ to all words under Pearson correlation: $\rho\left(\boldsymbol{\alpha}_{l',i}^{(m')}[h'], \boldsymbol{\alpha}_{l,i}^{(m)}[h]\right)$. Then average over words in the corpus.

- Jensen–Shannon divergence: for every word $x_i$, compare the attention distributions under Jensen–Shannon divergence: $\frac{1}{2}\,\mathrm{KL}(\alpha_{l',i}^{(m')}[h'] \,\|\, \beta) + \frac{1}{2}\,\mathrm{KL}(\alpha_{l,i}^{(m)}[h] \,\|\, \beta)$, where KL is the KL-divergence and $\beta$ is the average of the two attention distributions. Then average of words in the corpus.

As before, this gives rise to aggregate measures at the layer level by averaging over heads $h$.

## B Additional Representation-level Similarity Heatmaps

Figure 8 shows additional representation-level similarity heatmaps.

### B.1 Effect of Data Used for Similarity Measures

The majority of the experiments reported in the paper are using the Penn Treebank for calculating the similarity measures. Here we show that the results are consistent when using a different dataset, namely the Universal Dependencies English Web Treebank (Silveira et al., 2014). We repeat the experiment reported in Section 5.1. The resulting heatmaps, shown in Figure 9, are highly similar to those generated using the Penn Treebank, shown in Figure 8.

## C Additional Attention-level Similarity Heatmaps

Figure 10 shows additional attention-level similarity heatmaps.

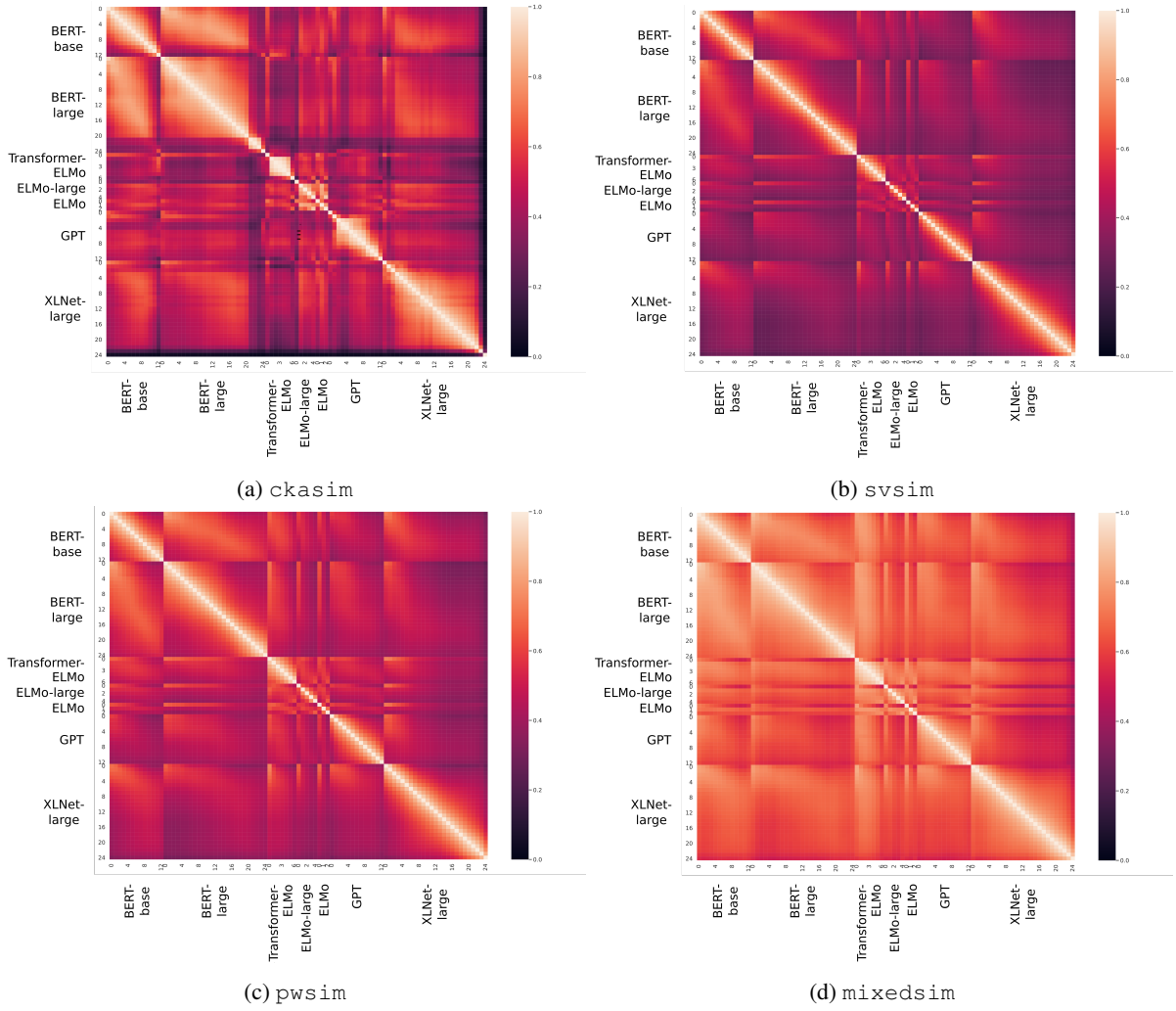
## D Efficient Fine-tuning

The analysis results showed that lower layers of the models go through limited changes during fine-tuning compared to higher layers. We use this insight to improve the efficiency of the fine-tuning process. In standard fine-tuning, back-propagation is done on the full network. We hypothesize that we can reduce the number of these operations by freezing the lower layers of the model since they are the least affected during the fine-tuning process. We experiment with freezing top and bottom layers of the network during the fine-tuning process. Different from prior work (Raghu et al., 2017; Felbo et al., 2017; Howard and Ruder, 2018), we freeze the selected layers for the complete fine-tuning process in contrast to freezing various layers for a fraction of the training time. We use the default parameters settings provided in the Transformer library (Wolf et al., 2019): batch size = 8, learning rate = $5e^{-5}$, Adam optimizer with epsilon = $1e^{-8}$, and number of epochs = 3.

| | Froze | SST-2 | MNLI | QNLI | QQP |
|---|---|---|---|---|---|
| | 0 | 92.43 | 84.05 | 91.40 | 91.00 |
| BERT | Top 4 | 91.86 | 82.86 | 91.09 | **90.97** |
| | Bot. 4 | **92.43** | **84.16** | **91.85** | 90.86 |
| | Top 6 | 91.97 | 82.53 | 90.13 | 90.61 |
| | Bot. 6 | **93.00** | **84.00** | **91.80** | **90.71** |
| | 0 | 93.92 | 85.97 | 90.35 | 90.55 |
| XLNet | Top 4 | 92.89 | 85.55 | 87.96 | **90.92** |
| | Bot. 4 | **93.12** | **86.04** | **90.65** | 89.36 |
| | Top 6 | 93.12 | 84.84 | 87.88 | **90.75** |
| | Bot. 6 | **93.92** | **85.64** | **90.99** | 89.02 |

Table 1: Freezing top/bottom 4/6 layers of BERT and XLNet during fine-tuning.

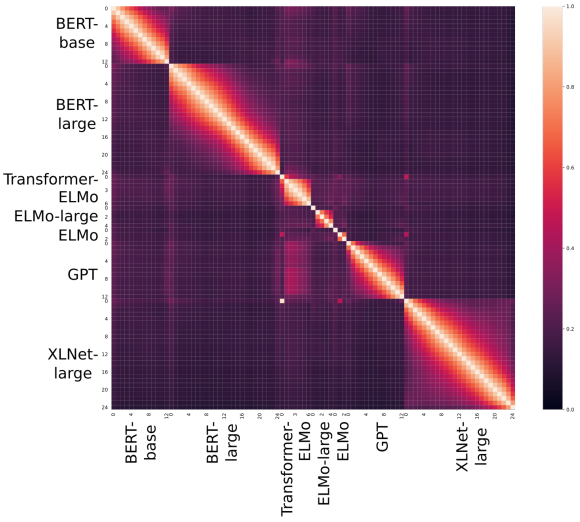
Table 1 presents the results on BERT and XLNet. On all of the tasks except QQP, freezing the bottom layers resulted in better performance than freezing the top layers. One interesting observation is that as we increase the number of bottom layers for freezing to six, the performance marginally degrades while saving a lot more computation. Surprisingly, on SST-2 and QNLI, freezing the bottom six layers resulted in better or equal performance than not freezing any layers of both models. With freezing the bottom six layers, one can save back-propagation computation by more than 50%.



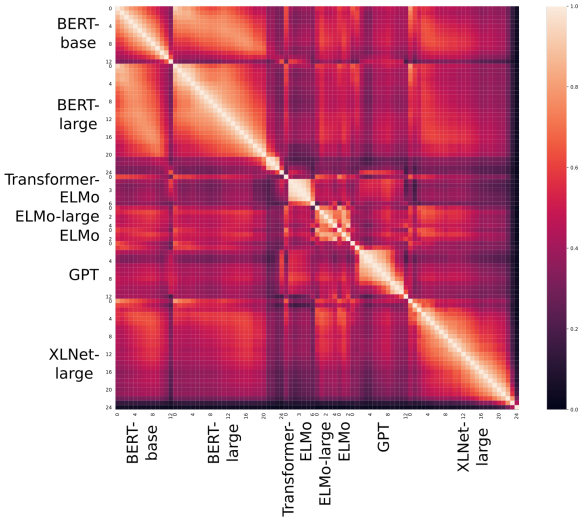

(a) `ckasim` (b) `svsim`

(c) `pwsim` (d) `mixedsim`

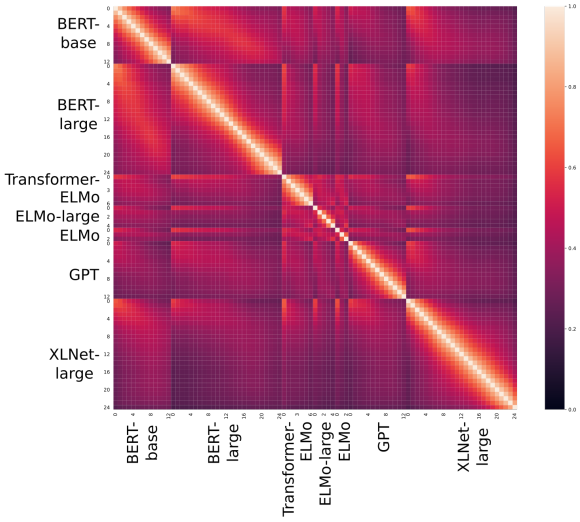
Figure 8: Similarity heatmaps of layers in various models under different representation-level similarity measures.



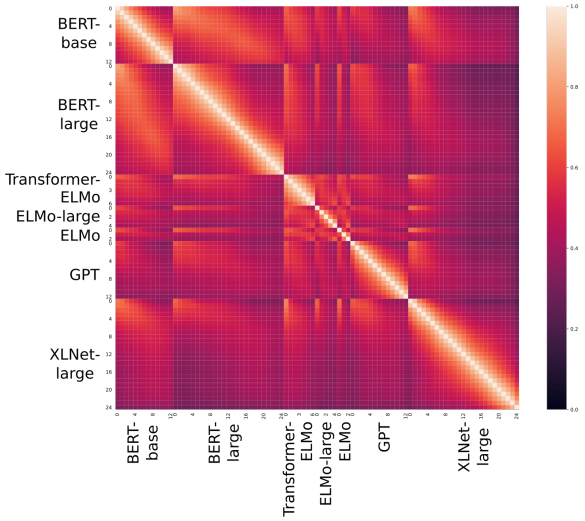

(a) `neuronsim`



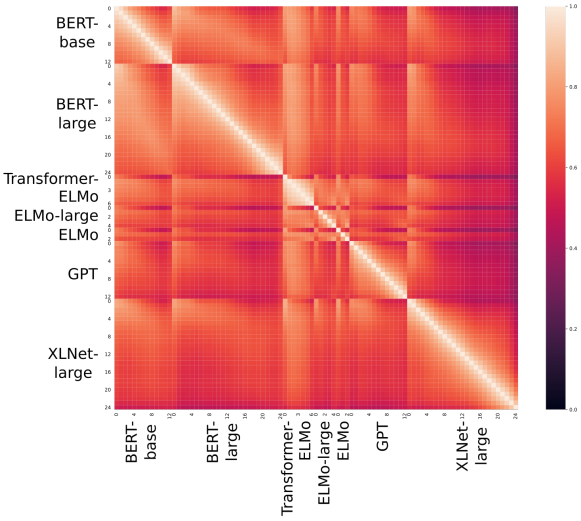

(b) `ckasim`




(c) `svsim`

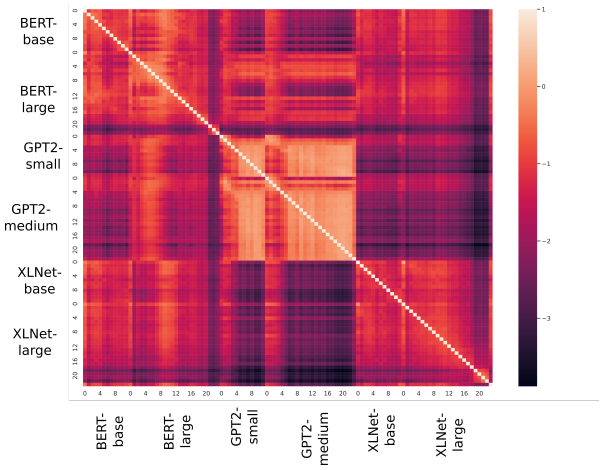



(d) `pwsim`

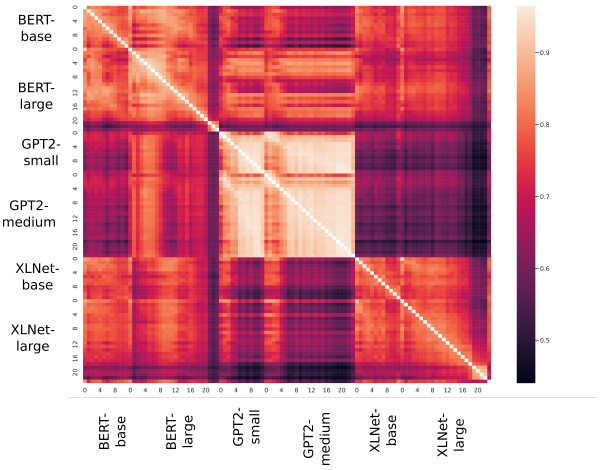



(e) `mixedsim`

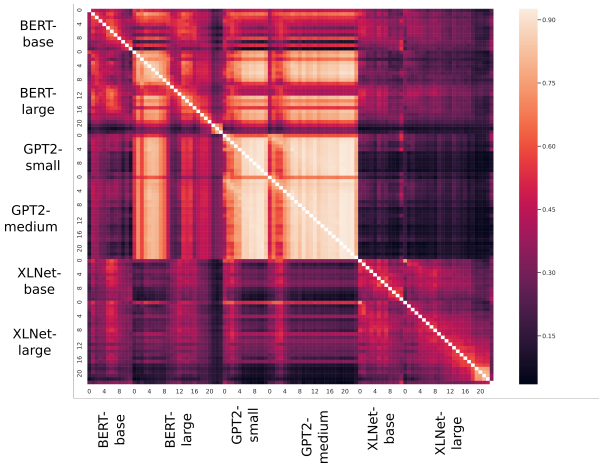
Figure 9: Similarity heatmaps of layers in various models under neuron-level and representation-level similarity measures, using the English Web Treebank corpus.



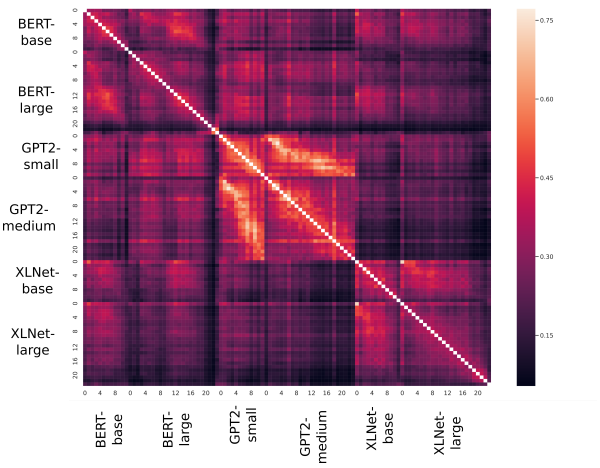

(a) Matrix norm



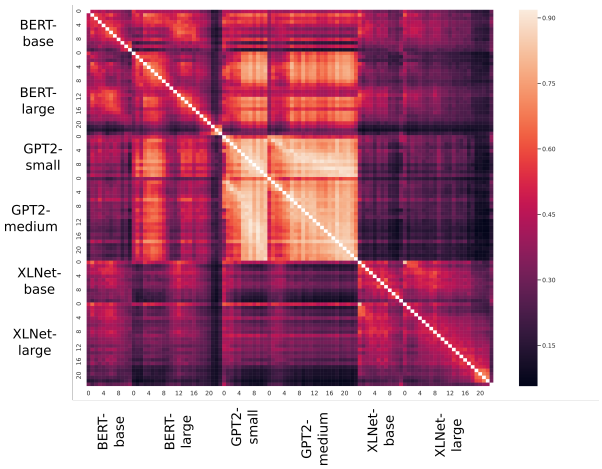

(b) Jensen–Shannon



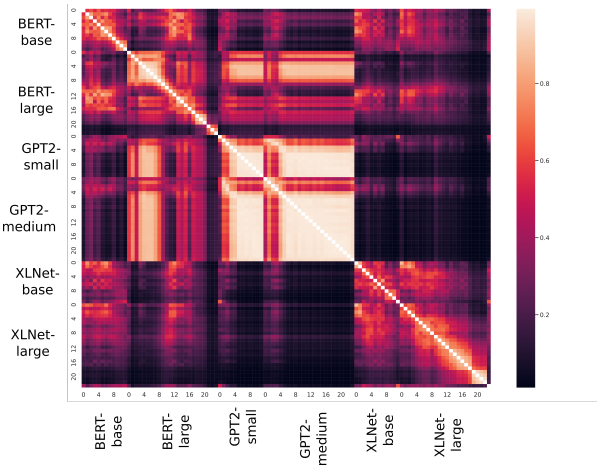

(c) Pearson




(d) `svsim`




(e) `pwsim`




(f) `ckasim`

Figure 10: Similarity heatmaps of layers in various models under different attention-level similarity measures.