Syntactic Perturbations Reveal Representational Correlates of Hierarchical Phrase Structure in Pretrained Language Models

Matteo Alleman[†] **Jonathan Mamou**[‡] **Miguel A Del Rio**[◇]
Hanlin Tang[‡] **Yoon Kim**[⋆,◇,•] **SueYeon Chung**[†,◇,•]

[†]Columbia University [‡]Intel Labs
[⋆]MIT-IBM Watson AI [◇]Massachusetts Institute of Technology

Abstract

While vector-based language representations from pretrained language models have set a new standard for many NLP tasks, there is not yet a complete accounting of their inner workings. In particular, it is not entirely clear what aspects of sentence-level syntax are captured by these representations, nor how (if at all) they are built along the stacked layers of the network. In this paper, we aim to address such questions with a general class of interventional, input perturbation-based analyses of representations from pretrained language models. Importing from computational and cognitive neuroscience the notion of representational invariance, we perform a series of probes designed to test the sensitivity of these representations to several kinds of structure in sentences. Each probe involves swapping words in a sentence and comparing the representations from perturbed sentences against the original. We experiment with three different perturbations: (1) random permutations of n-grams of varying width, to test the scale at which a representation is sensitive to word position; (2) swapping of two spans which do or do not form a syntactic phrase, to test sensitivity to global phrase structure; and (3) swapping of two adjacent words which do or do not break apart a syntactic phrase, to test sensitivity to local phrase structure. Results from these probes collectively suggest that Transformers build sensitivity to larger parts of the sentence along their layers, and that hierarchical phrase structure plays a role in this process. More broadly, our results also indicate that structured input perturbations widens the scope of analyses that can be performed on often-opaque deep learning systems, and can serve as a complement to existing tools (such as supervised linear probes) for interpreting complex black-box models.[1]

[1]Datasets, extracted features and code will be publicly available upon publication.

• Correspondence

1 Introduction

It is still unknown how distributed information processing systems encode and exploit complex relational structures in data, despite their ubiquitous use in the modern world. The fields of deep learning (Saxe et al., 2013; Hewitt and Manning, 2019), neuroscience (Sarafyazd and Jazayeri, 2019; Stachenfeld et al., 2017), and cognitive science (Elman, 1991; Kemp and Tenenbaum, 2008; Tervo et al., 2016) have given great attention to this question, including a productive focus on the potential models and their implementations of hierarchical tasks, such as predictive maps and graphs. In this work, we provide a generic means of identifying input structures that deep language models use to "chunk up" vastly complex data.

Natural (human) language provides a rich domain for studying how complex hierarchical structures are encoded in information processing systems. More so than other domains, human language is unique in that its underlying hierarchy has been extensively studied and theorized in linguistics, which provides source of "ground truth" structures for stimulus data. Much prior work on characterizing the types of linguistic information encoded in computational models of language such as neural networks has focused on supervised readout probes, which train a classifier on top pretrained models to predict a particular linguistic label (Belinkov and Glass, 2017; Liu et al., 2019a; Tenney et al., 2019). In particular, Hewitt and Manning (2019) apply probes to discover linear subspaces that encode tree-distances as distances in the representational subspace, and Kim et al. (2020) show that these distances can be used even without any labeled information to induce hierarchical structure. However, recent work has highlighted issues with correlating supervised probe performance with the amount of language structure encoded in such representations (Hewitt and Liang, 2019). Another popular approach to analyzing deep models is through the

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 263–276
Bangkok, Thailand (Online), August 6, 2021.

lens of geometry (Reif et al., 2019; Gigante et al., 2019). While geometric interpretations provide significant insights, they present another challenge in summarizing the structure in a quantifiable way. More recent techniques such as replica-based mean field manifold analysis method (Chung et al., 2018; Cohen et al., 2019; Mamou et al., 2020) connects representation geometry with linear classification performance, but the method is limited to categorization tasks.

In this work, we make use of an experimental framework from cognitive science and neuroscience to probe for hierarchical structure in contextual representations from pretrained Transformer models (i.e., BERT (Devlin et al., 2018) and its variants). A popular technique in neuroscience involves measuring change in the population activity in response to controlled, input perturbations (Mollica et al., 2020; Ding et al., 2016). We apply this approach to test the characteristic scale and the complexity (Fig. 1) of hierarchical phrase structure encoded deep contextual representations, and present several key findings:

1. Representations are distorted by shuffling small n-grams in early layers, while the distortion caused by shuffling large n-grams starts to occur in later layers, implying the scale of characteristic word length increases from input to downstream layers.

2. Representational distortion caused by swapping two constituent phrases is smaller than when the control sequences of the same length are swapped, indicating that the BERT representations are sensitive to hierarchical phrase structure.

3. Representational distortion caused by swapping adjacent words across phrasal boundary is larger than when the swap is within a phrasal boundary; furthermore, the amount of distortion increases with the syntactic distance between the swapped words. The correlation between distortion and tree distance increases across the layers, suggesting that the characteristic complexity of phrasal subtrees increases across the layers.

4. Early layers pay more attention between syntactically closer adjacent pairs and deeper layers pay more attention between syntactically distant adjacent pairs. The attention paid in each layer can explain some of the emergent sensitivity to phrasal structure across layers.

Our work demonstrates that interventional tools such as controlled input perturbations can be useful for analyzing deep networks, adding to the growing, interdisciplinary body of work which profitably adapt experimental techniques from cognitive neuroscience and psycholinguistics to analyze computational models of language (Futrell et al., 2018; Wilcox et al., 2019; Futrell et al., 2019; Ettinger, 2020).

2 Methods

Eliciting changes in behavioral and neural responses through controlled input perturbations is a common experimental technique in cognitive neuroscience and psycholinguistics (Tsao and Livingstone, 2008; Mollica et al., 2020). Inspired by these approaches, we perturb input sentences and measure the discrepancy between the resulting, perturbed representation against the original. While conceptually simple, this approach allows for a targeted analysis of internal representations obtained from different layers of deep models, and can suggest partial mechanisms by which such models are able to encode linguistic structure. We note that sentence perturbations have been primarily utilized in NLP for representation learning (Hill et al., 2016; Artetxe et al., 2018; Lample et al., 2018), data augmentation (Wang et al., 2018; Andreas, 2020), and testing for model robustness (e.g., against adversarial examples) (Jia and Liang, 2017; Belinkov and Bisk, 2018). A methodological contribution of our work is to show that input perturbations can complement existing tools and widens the scope of questions that could be asked of representations learned by deep networks.

2.1 Sentence perturbations

In this work we consider three different types of sentence perturbations designed to probe for different phenomena.

n-gram shuffling In the n-gram shuffling experiments, we randomly shuffle the words of a sentence in units of n-grams, with n varying from 1 (i.e., individual words) to 7 (see Fig. 2a for an example). While the number of words which change absolute position is similar for different n, larger n will better preserve the local context (i.e., relative position) of more words. Thus, we reason that n-gram swaps affect the representations selective to the context

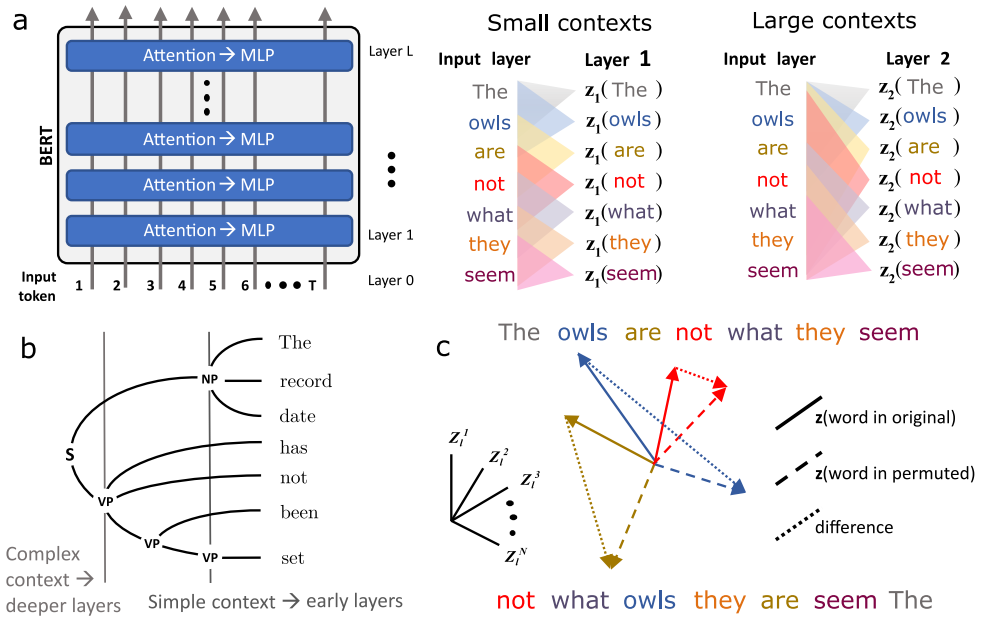

Figure 1: Do Transformers build complexity along their layers? (**a**) The representation of a word is a function of its context, and this cartoon illustrates an hypothesis that deeper representations use larger contexts. (**b**) An example parse tree, illustrating our notion of phrase complexity. (**c**) Cartoon of the distortion metric, where vectors are the z-scored feature vectors $\mathbf{z}$, and color map vectors to words.

with size n or higher within the sentence, and that lower n will result in greater distortion in sentence representations.

Phrase swaps The n-gram shuffling experiments probe for sensitivity of representations to local context without taking into account syntactic structure. In the phrase swap experiments, we perturb a sentence by swapping two randomly chosen spans. We explore two ways of swapping spans. In the first setting, the spans are chosen such that they are valid phrases according to its parse tree.[3] In the second setting, the spans are chosen that they are invalid phrases. Importantly, in the second, control setting, we fix the length of the spans such that the lengths of spans that are chosen to be swapped are the same as in the first setting (see Fig. 3a for an example). We hypothesize that swapping invalid phrases will result in more distortion than swapping valid phrases, since invalid swaps will result in greater denigration of syntactic structure.

Adjacent word swaps In the adjacent word swapping experiments, we swap two adjacent words in a sentence. We again experiment with two settings – in the first setting, the swapped words stay within the phrase boundary (i.e., the two words share the same parent), while in the second setting, the swapped words cross phrase boundaries. We also perform a more fine-grained analysis where we condition the swaps based on the "syntactic distance" between the swapped words, where syntactic distance is defined as the distance between the two words in the parse tree (see Fig. 6c). Since a phrase corresponds to a subtree of the parse tree, this distance also quantifies the number of nested phrase boundaries between two adjacent words. Here, we expect the amount of distortion to be positively correlated with the syntactic distance of the words that are swapped.

[3]We use constituency parse trees from the English Penn Treebank (Marcus et al., 1994).

2.2 Contextual representations from Transformers

For our sentence representation, we focus on the Transformer-family of models pretrained on large-scale language datasets (BERT and its variants). Given an input word embedding matrix $\mathbf{X} \in \mathbb{R}^{T \times d}$ for a sentence of length T, the Transformer applies self attention over the previous layer's representation to produce a new representation,

$$\mathbf{X}_l = f_l([\mathbf{H}_{l,1}, \ldots, \mathbf{H}_{l,H}]), \quad \mathbf{H}_{l,i} = \mathbf{A}_{l,i}\mathbf{X}_{l-1}\mathbf{V}_{l,i},$$
$$\mathbf{A}_{l,i} = \text{softmax}\left(\frac{(\mathbf{X}_{l-1}\mathbf{Q}_{l,i})(\mathbf{X}_{l-1}\mathbf{K}_{l,i})^\top}{\sqrt{d_k}}\right), \quad (1)$$

where f_l is an MLP layer, H is the number of heads, $d_H = \frac{d}{H}$ is the head embedding dimension, and $\mathbf{Q}_{l,i}, \mathbf{K}_{l,i}, \mathbf{V}_{l,i} \in \mathbb{R}^{d \times d_k}$ are respectively the learned query, key, and value projection matrices at layer l for head i. The MLP layer consists of a residual layer followed by layer normalization and a nonlinearity. The 0-th layer representation

$\mathbf{X}_0$ is obtained by adding the position embeddings and the segment embeddings to the input token embeddings $\mathbf{X}$, and passing it through normalization layer.[4]

In this paper, we conduct our distortion analysis mainly on the intermediate Transformer representations $\mathbf{X}_l = [\mathbf{x}_{l,1}, \ldots, \mathbf{x}_{l,T}]$, where $\mathbf{x}_{l,t} \in \mathbb{R}^d$ is the contextualized representation for word t at layer l.[5] We analyze the trend in distortion as a function of layer depth l for the different perturbations. We also explore the different attention heads $\mathbf{H}_{l,i} \in \mathbb{R}^{T \times d_H}$ and the associated attention matrix $\mathbf{A}_{l,i} \in \mathbb{R}^{T \times T}$ to inspect whether certain attention heads specialize at encoding syntactic information.

2.3 Distortion metric

Our input manipulations allow us to specify the distortion at the input level, and we wish to measure the corresponding distortion in the representation space (Fig. 1). Due to the attention mechanism, a single vector in an intermediate layer is a function of the representations of (potentially all) the other tokens in the sentence. Therefore, the information about a particular word might be distributed among the many feature vectors of a sentence, and we wish to consider all feature vectors together as a single sentence-level representation.

We thus represent each sentence as a matrix and use the distance induced by matrix 2-norm. Specifically, let $\mathbf{P} \in \{0,1\}^{T \times T}$ be the binary matrix representation of a permutation that perturbs the input sentence, i.e., $\tilde{\mathbf{X}} = \mathbf{P}\mathbf{X}$. Further let $\tilde{\mathbf{X}}_l$ and $\mathbf{X}_l$ be the corresponding sentence representations for the l-th layer for the perturbed and original sentences. To ignore uniform shifting and scaling, we also z-score each feature dimension of each layer (by subtracting the mean and dividing by the standard deviation where these statistics are obtained from the full Penn Treebank corpus) to give $\tilde{\mathbf{Z}}_l$ and $\mathbf{Z}_l$. Our distortion metric for layer l is then defined as $\|\mathbf{Z}_l - \mathbf{P}^{-1}\tilde{\mathbf{Z}}_l\|/\sqrt{Td}$, where $\|\cdot\|$ is the matrix 2-norm (i.e., Frobenius norm).[6] Importantly, we invert the permutation of the perturbed representation to preserve the original ordering, which allows us to measure the distortion due to structural change, rather than distortion due to simple differences in surface form. We divide by $\sqrt{Td}$ to make the metric comparable between sentences (with different T) and networks (with different d).

Intuitively, our metric is the scaled Euclidean distance between the z-scored, flattened sentence representations, $\mathbf{z}_l \in \mathbb{R}^{Td}$. Because each dimension is independently centered and standardized, the maximally unstructured distribution of $\mathbf{z}_l$ is an isotropic Td-dimensional Gaussian. The expected distance between two such vectors is approximately $\sqrt{2Td}$. Therefore, we can interpret a distortion value approaching $\sqrt{2}$ as comparable to if we had randomly redrawn the perturbed representation.

3 Experimental Setup

We apply our perturbation-based analysis on sentences from the English Penn Treebank (Marcus et al., 1994), where we average the distortion metric across randomly chosen sentences. We analyze the distortion, as measured by length-normalized Frobenius norm between the perturbed and original representations, as a function of layer depth. Layers that experience large distortion when the syntactic structure is disrupted from the perturbation can be interpreted as being more sensitive to hierarchical syntactic structure.

As we found the trend to be largely similar across the different models, in the following section, we primarily discuss results from BERT (`bert-base-cased`). We replicate key results with other pretrained and randomly-initialized Transformer-based models as well (see A.1).

4 Results

4.1 Sensitivity to perturbation size increases along BERT layers

When we shuffle in units of larger n-grams, it only introduces distortions in the deeper BERT layers compared to smaller n-gram shuffles. The n-gram sized shuffles break contexts larger than n, while preserving contexts of size n or smaller. Interestingly, smaller n-gram shuffles diverge from the original sentence in the early layers (Fig. 2b, top curve), implying that only in early layers are representations built from short-range contexts. Larger n-gram shuffles remain minimally distorted for 'longer' (Fig. 2b, bottom curve), implying that long-

[4] However, the exact specification for the MLP and $\mathbf{X}_0$ may vary across different pretrained models.

[5] BERT uses BPE tokenization (Sennrich et al., 2015), which means that some words are split into multiple tokens. Since we wish to evaluate representations at word-level, if a word is split into multiple tokens, its word representation is computed as the average of all its token representations.

[6] There are many possible ways of measuring distortion, induced by different norms. We observed the results to be qualitatively similar for different measures, and hence we focus on the Frobenius norm in our main results. We show the results from additional distortion metrics in the A.2

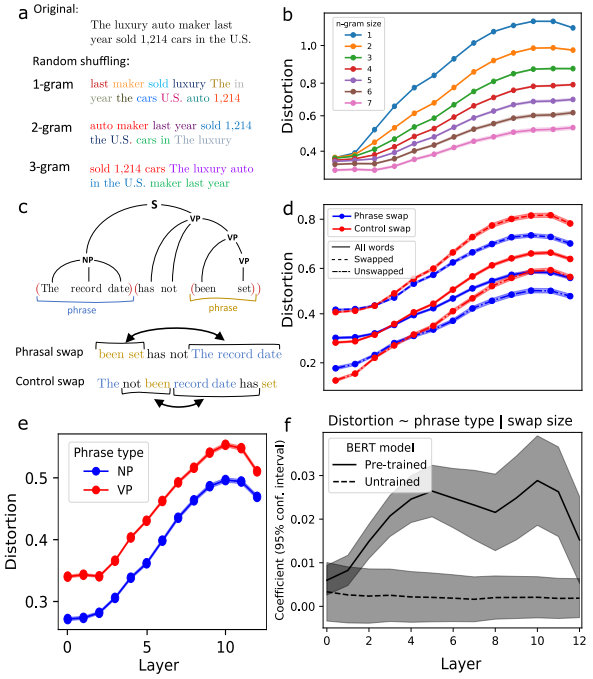

Figure 2: Swapping n-grams and phrases. (**a**) Examples of basic n-gram shuffles, where colors indicate the units of shuffling. (**b**) Distortion metric computed at each layer, conditioned on n-gram size. Error bars hereafter represent standard error across 400 examples. (**c**) An example parse tree, with phrase boundaries shown as grey brackets, and two low-order phrases marked; and examples of a phrasal and control swap, with colors corresponding to the phrases marked above. (**d**) Distortion, computed at each layer, using either the full sentence, the subsentence of unswapped words, or the subsentence of swapped words, conditioned on swap type. (e) Full-sentence distortion for VP and NP phrase swaps. (f) Partial linear regression coefficients (see A.4) for pre-trained and untrained BERT models after controlling for swap size.

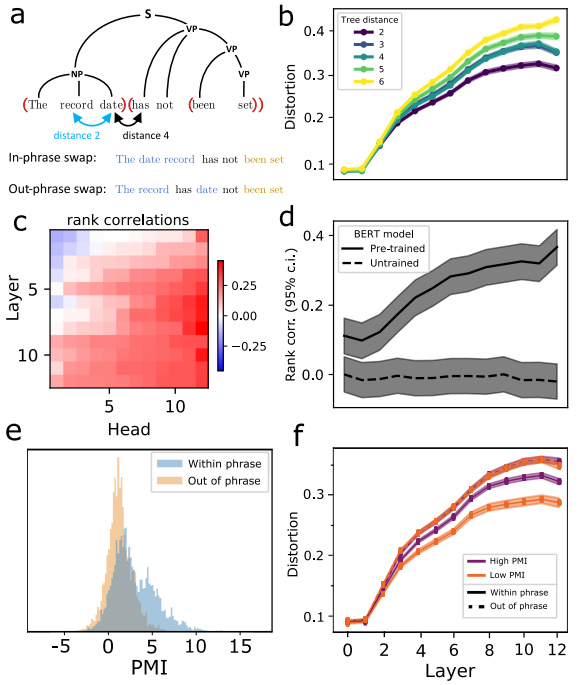

Figure 3: Syntactic distance affects representational distortion. (**a**) An example of adjacent swaps which do and do not cross a phrase boundary, with low-order phrases colored. Phrase boundaries are drawn in red. (**b**) Distortion in each layer, but conditioned on the tree distance. (**c**) For each head (column) of each layer (row), the (Spearman) rank correlation between distortion and tree distance of the swapped words. Colors are such that red is positive, blue negative. (**d**) Rank correlations between distortion (of the full representation) in the trained and untrained BERT models. (**e**) Histogram of PMI values, for pairs in the same phrase and not. (**f**) Similar to **b**, but averaging all out-of-phrase swaps, and separating pairs above ('high') or below ('low') the median PMI.

range contexts play a larger role deeper layer representations.

Effects of phrasal boundaries Since BERT seems to build larger contexts along its layers, we now ask whether those contexts are structures of some grammatical significance. A basic and important syntactic feature is the constituent phrase, which BERT has previously been shown to represented in some fashion (Goldberg, 2019; Kim et al., 2020). We applied two targeted probes of phrase structure in the BERT representation, and found that phrasal boundaries are indeed influential.

If we swap just two n-grams, the BERT representations are less affected when phrases are kept intact. We show this by swapping only two n-grams per sentence and comparing the distortion when those n-grams are phrases to when they cross phrase boundaries (Fig. 3a), where we control for the length of n-grams that are swapped in both settings. There is less distortion when respecting phrase boundaries, which is evident among all feature vectors, including those in the position of words which did not get swapped (Fig. 2d). The global contextual information, distributed across the sentence, is affected by the phrase boundary.

To see if the role of a phrase impacts its salience, we distinguish between verb phrases (VP) and noun phrase (NP) swaps. Swapping VP results in more distortion than swapping NP (Fig. 2e). Since VP are in general larger than NP, this effect could in principle be due simply to the number of words being swapped. Yet that is not the case: Using a partial linear regression (see details in A.4), we can estimate the difference between the VP and NP distortions conditional on any smooth function of the swap size, and doing this reveals that there is still a strong difference in the intermediate layers (Fig. 2f).

4.2 Sensitivity depends on syntactic distance of the perturbation

Having seen that representations are sensitive to phrase boundaries, we next explore whether that

sensitivity is proportional to the number of phrase boundaries that are broken, which is a quantity related to the phrase hierarchy. Instead of swapping entire phrases, we swap two adjacent words and analyze the distortion based on how far apart the two words are in the constituency tree (Fig. 3a)[7]. This analysis varies the distance in the deeper tree structure while keeping the distance in surface form constant (since we always swap adjacent words).

If the hierarchical representations are indeed being gradually built up along the layers of these pretrained models, we expect distortion to be greater for word swaps that are further apart in tree distance. We indeed find that there is a larger distortion when swapping syntactically distant words (Fig. 3b). This distortion grows from earlier to later BERT layers. Furthermore, when looking at the per-head representations of each layer, we see that in deeper layers there are more heads showing a positive rank correlation between distortion and tree distance (Fig. 3c). In addition to a sensitivity to phrase boundaries, deeper BERT layers develop a sensitivity to the number of boundaries that are broken.

Controlling for co-occurrence Since words in the same phrase may tend to occur together more often, co-occurrence is a potential confound when assessing the effects of adjacent word swaps. Co-occurrence is a simple statistic which does not require any notion of grammar to compute – indeed it is used to train many non-contextual word embeddings (e.g., word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014)). So it is natural to ask whether BERT's resilience to syntactically closer swaps goes beyond simple co-occurrence statistics. For simplicity, let us focus on whether a swap occurs within a phrase (tree distance = 2) or not.

As an estimate of co-occurrence, we used the pointwise mutual information (PMI). Specifically, for two words w and v, the PMI is $\log \frac{p(w,v)}{p(w)p(v)}$, which is estimated from the empirical probabilities. We confirm that adjacent words in the same phrase do indeed have a second mode at high PMI (Fig. 3e). Dividing the swaps into those whose words have high PMI (above the marginal median) and low PMI (below it), we can see visually that the difference between within-phrase swaps and out-of-phrase swaps persists in both groups (Fig. 3f). When quantitatively accounting for the effect of PMI with a partial linear regression (see A.4), there remains a significant correlation between the breaking of a phrase and the subsequent distortion. This indicates that the greater distortion for word swaps which cross phrase boundaries is not simply due to surface co-occurrence statistics.

Relation to linguistic information Do our input perturbations, and the resulting the distortions, reflect changes in the encoding of important linguistic information? One way to address this question, which is popular in computational neuroscience (DiCarlo and Cox, 2007) and more recently BERTology (Liu et al., 2019a; Tenney et al., 2019), is to see how well a linear classifier trained on a linguistic task generalizes from the (representations of the) unperturbed sentences to the perturbed ones. With supervised probes, we can see how much the representations change with respect to the subspaces that encode specific linguistic information.

Specifically, we relate representational distortion to three common linguistic tasks of increasing complexity: part of speech (POS) classification; grandparent tag (GP) classification (Tenney et al., 2019); and a parse tree distance reconstruction (Hewitt and Manning, 2019)[8]. The probe trained on each of these tasks is a generalized linear model, mapping a datapoint $\mathbf{x}$ (i.e. representations from different layers) to a conditional distribution of the labels, $p(y|\theta^T\mathbf{x})$ (see A.5 for model details). Thus a ready measure of the effect of each type of swap, for a single sentence, is $\log p(y|\theta^T\mathbf{x}_i) - \log p(y|\theta^T\tilde{\mathbf{x}}_i)$, where $\tilde{\mathbf{x}}_i$ is same datum as $\mathbf{x}_i$ in the perturbed representation[9]. Averaging this quantity over all datapoints gives a measure of content-specific distortion within a representation, which we will call "inference impairment".

Based on the three linguistic tasks, the distortion we measure from the adjacent word swaps is more strongly related to more complex information. The inverted L shape of Fig. 4a suggests that increasing distortion is only weakly related to impairment of POS inference, which is perhaps unsurprising given that POS tags can be readily predicted from

[7] Note that for adjacent words, the number of broken phrase boundaries equals the tree distance minus two.

[8] While the original paper predicted dependency tree distance, in this paper we instead predict the constituency tree distance.

[9] POS- and GP-tag prediction outputs a sequence of labels for each sentence, while the distance probe outputs the constituency tree distance between each pair of words. Then $\log p(y|\theta^T\mathbf{x}_i)$ is simply the log probability of an individual label.

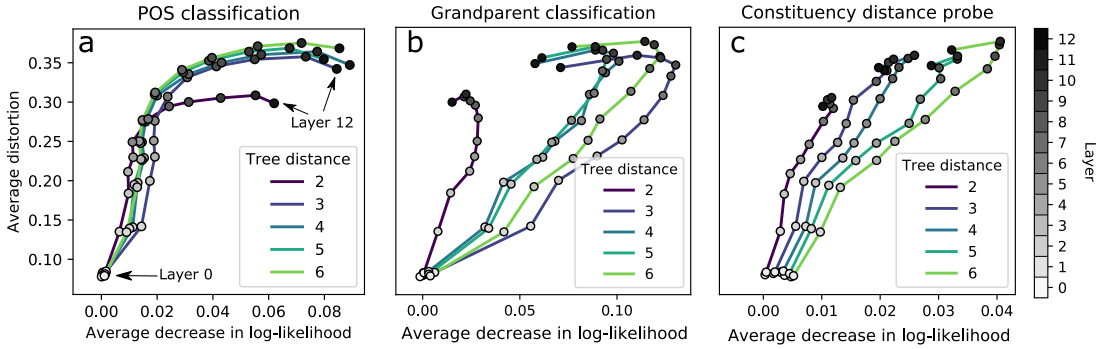

Figure 4: Distortion and inference impairment for increasing linguistic complexity. In each plot, a point is the average (distortion, 'impairment') for a given layer and a given class of word swap distance. Points are connected by lines according to their swap type (i.e. tree distance). The circles are colored according to layer (see right for a legend). Averages are taken over 600 test sentences, with one of each swap type per sentence, and both distortion and log-likelihood are computed for every word in the sentence.

local context. A deeper syntactic probe, the GP classifier (4b), does show a consistent positive relationship, but only for swaps which break a phrase boundary (i.e. distance >2). Meanwhile, impairment of the distance probe (4c), which reconstructs the full parse tree, has a consistently positive relationship with distortion, whose slope is proportionate to the tree distance of the swap. Thus, when specifically perturbing the phrasal boundary, the representational distortion is related to relatively more complex linguistic information.

4.3 Sensitivity to perturbations is mediated by changes in attention

In the transformer architecture, contexts are built with the attention mechanism. Recall that attention is a mechanism for allowing input vectors to interact when forming the output, and the ultimate output for a given token is a convex combination of the features of all tokens (Eq. 1). It has been shown qualitatively that, within a layer, BERT allocates attention preferentially to words in the same phrase (Kim et al., 2020), so if our perturbations affect inference of phrase structure then the changes in attentions could explain our results. Note that it is not guaranteed to do so: the BERT features in a given layer are a function of the attentions and the "values" (each token's feature vector), and both are affected by our perturbations. Therefore our last set of experiments asks whether attention alone can explain the sensitivity to syntactic distance.

To quantify the change in attention weights across the whole sentence, we compute the distance between each token's attention weights in the perturbed and unperturbed sentences, and average across all tokens. For token i, its vector of attention weights in response to the unperturbed sentence is a^i, and for the perturbed one $\tilde{a}^i$ (such that $\sum_j a^i_j = 1$). Since each set of attention weights are non-negative and sum to 1 due to softmax, we use the relative entropy[10] as a distance measure. This results in the total change in attention being:

$$\Delta a = \frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{T} a^i_j \log \frac{a^i_j}{\tilde{a}^i_j}$$

which is non-negative and respects the structure of the weights. We confirmed that other measures (like the cosine similarity) produce results that are qualitatively similar.

First, we observe that the changes in the attention depend on the layer hierarchy when adjacent word swaps break the phrase boundary. Like the distortion, attention changes little or not at all in the early layers, and progressively more in the final layers (Fig. 5b). Furthermore, these changes are also positively correlated with syntactic distance in most cases, which suggests that representation's sensitivities to syntactic tree distance may primarily be due to changes in attention.

To see whether the changes in attention can in fact explain representational sensitivity to syntactic distance, we turned to the same partial linear regression model as before (A.4) to compute the the correlation between the representation's distortion and the tree distance between the swapped adjacent words, after controlling for changes in attention (Δa). The correlations substantially reduced in the controlled case (Fig. 5c), which suggests that attention weights contribute to the representational sensitivity to syntactic tree distance; but the correlations are not eliminated, which suggests that distortions from the previous layer also contribute.

[10] Also called the KL divergence.

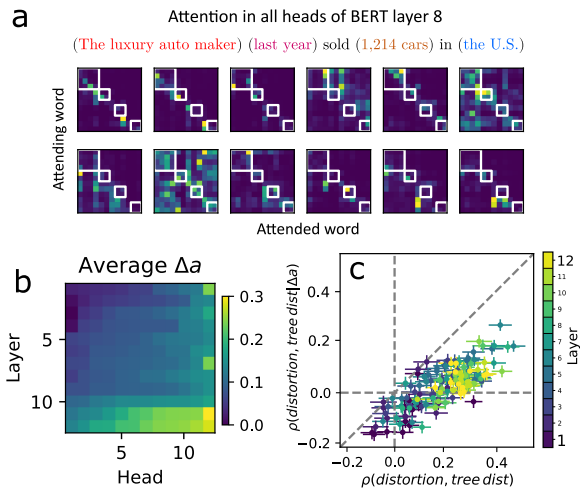

Figure 5: Attention changes explain part of the sensitivity to tree distance. (**a**) An example of the attention matrices for all heads in a single layer (layer 8), given the above sentence as input. Phrases in the sentence are drawn as blocks in the matrix. (**b**) The change in attention between the unperturbed and perturbed attention weights, averaged over all out-of-phrase swaps. Columns are sorted independently by their value. (**c**) The head/layer-wise rank correlations ($\pm 95\%$ confidence intervals) between distortion and tree distance after controlling for changes in attention, plotted against the uncontrolled rank correlations. Being below the diagonal indicates that the relationship between distortion and tree distance is partially explained by Δa.

5 Discussion and Conclusion

In this paper, we used the representational sensitivity to controlled input perturbation as a probe of hierarchical phrasal structure in deep linguistic representations. The logic of our probe is the representations which respect phrase structure should less sensitive to perturbations which preserve the phrasal unit, and more sensitive to those which disrupt a phrase. We hope that our results demonstrate the versatility and utility of perturbation-based approaches to studying deep language models.

We showed that BERT and its variants build representations which are sensitive to the phrasal unit, as demonstrated by greater invariance to perturbations preserving phrasal boundaries compared to control perturbations which break the phrasal boundaries (Fig. 2-5). We also find that while the representational sensitivity to broken phrase boundaries grows across layers, this increase in sensitivity is more prominent when the breakage occurs between two words that are syntactically distant (i.e., when the broken phrase is more complex). Using the same methods to show that changes in attention provide a partial explanation for perturbation-induced distortions.

While our distortion metric is a task-agnostic measure of change in the neural population activity, this may or may not reflect changes in the encoding of specific linguistic information. To relate our metric with specific kinds of information, we measured the change in the performance of supervised linear probes trained on top of the representation (Fig. 4). The probe sensitivity measure also bears a suggestive resemblance to the saliency map analysis (Simonyan et al., 2014) in machine learning, which is used to highlight the most output-sensitive regions within the input. To draw an analogy with that work, one way of characterizing our results is that phrasal boundaries are regions of high saliency in hidden representations and that, in deep layers, complex phrase boundaries are more salient than simple phrase boundaries. Further exploring the use of supervised probes and our input perturbations as a tool for layerwise probing of syntactic saliency is a promising direction for future work.

Finally, several studies (Sinha et al., 2021; Gupta et al., 2021; Pham et al., 2020), have recently found that masked language models pretrained or fine-tuned on sentences that break natural word order (e.g. via n-gram shuffling) still perform quite well across various tasks, even on supervised probes of syntactic phenomena. It would be interesting to apply our perturbative analyses on such models to see if they exhibit less sensitivity to the experimental vs. control setups (e.g. n-gram vs. phrase swaps). This may indicate that such models do not capture representational correlates of phrase structure in their representations despite their good performance on supervised probing tasks. In such a case, what tasks would actually require the "linguistic knowledge" that we are probing for? In similar vein, applying our perturbative analyses on models that explicitly incorporate syntax into their representations (Sundararaman et al., 2019; Wang et al.; Zanzotto et al., 2020; Kuncoro et al., 2020) might provide further insights.

Our method and results suggest many interesting future directions. We hope that this work will motivate: (1) a formal theory of efficient hierarchical data representations in distributed features; (2) a search for the causal connection between attention structure, the representational geometry, and the model performance; (3) potential applications in network pruning studies; (4) an extension of the current work as a hypothesis generator in neuroscience to understand how neural populations implement tasks with an underlying compositional structure.

References

Jacob Andreas. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings ACL*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of ICLR*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of ICLR*.

Yonatan Belinkov and James Glass. 2017. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Advances in Neural Information Processing Systems*, pages 2441–2451.

SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. 2018. Classification and geometry of general perceptual manifolds. *Physical Review X*, 8(3):031003.

Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. 2019. Separability and geometry of object manifolds in deep neural networks. *bioRxiv*, page 644658.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

James J DiCarlo and David D Cox. 2007. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341.

Nai Ding, Lucia Melloni, Hang Zhang, Xing Tian, and David Poeppel. 2016. Cortical tracking of hierarchical linguistic structures in connected speech. *Nature neuroscience*, 19(1):158.

Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3):195–225.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Richard Futrell, Ethan Wilcox, Takashi Morit, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv:1809.01329*.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Scott Gigante, Adam S Charles, Smita Krishnaswamy, and Gal Mishne. 2019. Visualizing the phate of neural networks. In *Advances in Neural Information Processing Systems*, pages 1840–1851.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert family eat word salad: Experiments with text understanding. *AAAI*.

Bruce Hansen. 2000. *Econometrics*. https://www.ssc.wisc.edu/~bhansen/econometrics/.

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings of EMNLP*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word 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 4129–4138.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*, Copenhagen, Denmark. Association for Computational Linguistics.

Charles Kemp and Joshua B Tenenbaum. 2008. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang goo Lee. 2020. Are Pre-trained Language Models Aware of Phrases? Simple but Strong Baselines for Grammar Induction. In *Proceedings of ICLR*.

Adhiguna Kuncoro, Lingpeng Kong, Daniel Fried, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. 2020. Syntactic Structure Distillation Pretraining for Bidirectional Encoders. *Transactions of the Association for Computational Linguistics*, 8:776–794.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of ICLR*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. *CoRR*, abs/1903.08855.

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.

Jonathan Mamou, Hang Le, Miguel A Del Rio, Cory Stephenson, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2020. Emergence of separable manifolds in deep language representations. *arXiv preprint arXiv:2006.01095*.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. 2020. Composition is the core driver of the language-selective network. *Neurobiology of Language*, 1(1):104–134.

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 5732–5741. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?

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 6076–6085. Curran Associates, Inc.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, pages 8592–8600.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Morteza Sarafyazd and Mehrdad Jazayeri. 2019. Hierarchical reasoning by neural circuits in the frontal cortex. *Science*, 364(6441):eaav8911.

Andrew M Saxe, James L McClellans, and Surya Ganguli. 2013. Learning hierarchical categories in deep neural networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. 2017. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2019. Syntax-Infused Transformer and BERT models for Machine Translation and Natural Language Understanding. *arXiv:1911.06156*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of ACL*.

D Gowanlock R Tervo, Joshua B Tenenbaum, and Samuel J Gershman. 2016. Toward the neural implementation of structure learning. *Current opinion in neurobiology*, 37:99–105.

Doris Y Tsao and Margaret S Livingstone. 2008. Mechanisms of face perception. *Annu. Rev. Neurosci.*, 31:411–437.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of EMNLP*.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019. Hierarchical representation in neural language models: Suppression and recovery of expectations. *arXiv preprint arXiv:1906.04068*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of NeurIPS*.

Fabio Massimo Zanzotto, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. 2020. KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A Appendix

Here we go into further detail on our methods and data to aid in reproducibility.

A.1 Model details

Here we give the details for all models considered in this paper. The majority of results are from BERT, but we also tested other variants.[11]

- **BERT** (Devlin et al., 2018) `bert-base-cased`. 12-layer, 768-hidden, 12-heads, 110M parameters.
- **RoBERTa** (Liu et al., 2019b) `roberta-base`. 12-layer, 768-hidden, 12-heads, 125M parameters.
- **ALBERT** (Lan et al., 2019) `albert-base-v1`. 12 repeating layers, 128 embedding, 768-hidden, 12-heads, 11M parameters.
- **DistilBERT** (Sanh et al., 2019) `distilbert-uncased`. 6-layer, 768-hidden, 12-heads, 66M parameters. The model distilled from the BERT model `bert-base-uncased` checkpoint.
- **XLNet** (Yang et al., 2019) `xlnet-base-cased`. 12-layer, 768-hidden, 12-heads, 110M parameters.

Note that the hidden size is 768 across all the models. For each pre-trained model, input text is tokenized using its default tokenizer and features are extracted at token level.

[11] We use the implementation from https://github.com/huggingface/transformers.

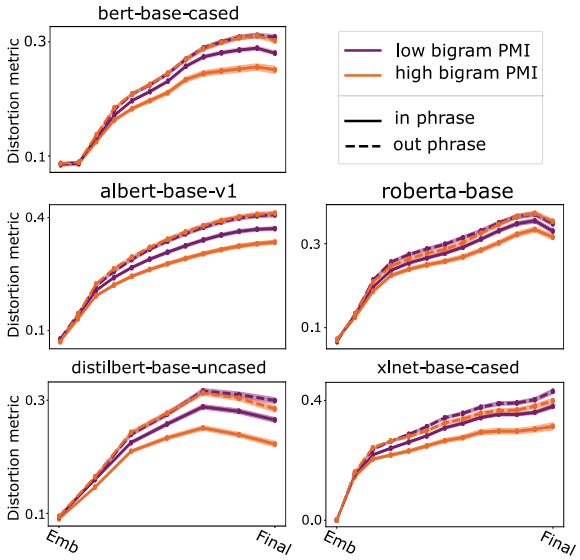

Figure 6: Replicating the adjacent word swapping experiments using different transformer architectures. Lines are the mean Frobenius distance, and the shading is ± 1 standard error of the mean.

A.2 Additional metrics

In addition to the scaled Frobenius distance, we also considered other ways of measuring distortion in the representation. We will briefly report results for two other metrics, and describe them here.

CCA Canonical correlations analysis (CCA) (Raghu et al., 2017) measures the similarity of two sets of variables using many samples from each. Given two sets of random variables $\mathbf{x} = (x_1, x_2, ..., x_n)$ and $\mathbf{y} = (y_1, y_2, ..., y_m)$, CCA finds linear weights $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ which maximise $\text{cov}(\mathbf{a} \cdot \mathbf{x}, \mathbf{b} \cdot \mathbf{y})$. In our context, we treat the representation of the original sentence as $\mathbf{x}$, and the representation of the perturbed sentence as $\mathbf{y}$, and the resulting correlation as a similarity measure.

Since CCA requires many samples, we use the set of all word-level representations across all perturbed sentences. For example, to construct the samples of $\mathbf{x}$ from S perturbed sentences, we get use $[\mathbf{X}_1|\mathbf{X}_2|...|\mathbf{X}_S]$, where each $\mathbf{X}_i \in \mathbb{R}^{768 \times T_i}$. Unless specified otherwise, $S = 400$. For good estimates, CCA requires many samples (on the order of at least the number of dimensions), and we facilitate this by first reducing the dimension of the matrices using PCA. Using 400 components preserves $\sim 90\%$ of the variance. Thus, while CCA gives a good principled measure of representational similarity, its hunger for samples makes it unsuitable as a per-sentence metric.

We also measured distortion using Projec-

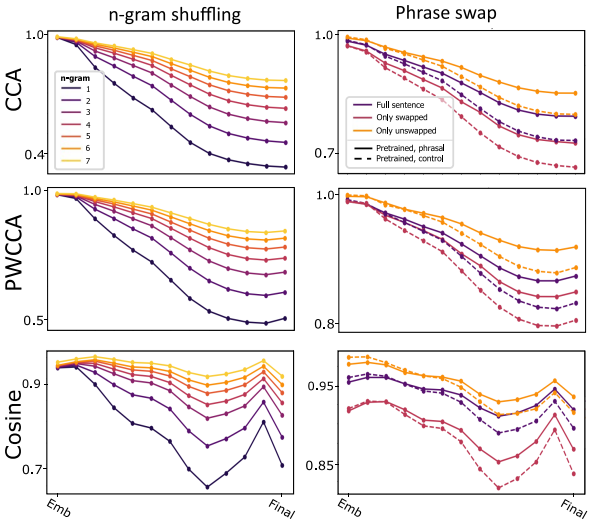

Figure 7: Results from the pretrained BERT model using alternative distortion metrics, on the n-gram shuffling and phrase swap experiments.

tion Weighted Canonical Correlation Analysis (PWCCA), an improved version of CCA to estimate the true correlation between tensors (Morcos et al., 2018).[12]

As reported in Figure 7, we did not find any qualitative differences between PWCCA and CCA in our experiments.

Cosine A similarity measure defined on individual sentences is the cosine between the sentence-level representations. By sentence-level representation, we mean the concatenation of the word-level vectors into a single vector $\mathbf{s} \in \mathbb{R}^{NT}$ (where N is the dimension of each feature vector). Treating each dimension of the vector as a sample, we can then define the following metric: $\text{corr}\left(\mathbf{s}_i^{original}, \mathbf{s}_i^{swapped}\right)$. This is equivalent to computing the cosine of the vectors after subtracting the (scalar) mean across dimensions, hence we will refer to it as 'cosine'.

A.3 Additional details on the dataset

In this section, we describe additional details of the manipulations done on the datasets.

n-gram shuffling For a given a sentence, we split it into sequential non-overlapping n-gram's from left to right; if the length of the sentence is not a multiple of n, the remaining words form an additional m-gram, $m < n$. The list of the n-gram's is randomly shuffled. Note that the 1-gram case is equivalent to a random shuffling of the words.

[12]For both CCA and PWCCA, we use the implementation from https://github.com/google/svcca.

In our analysis, we consider n-grams, with n varying from 1 (i.e., individual words) to 7 and all the sentences have at least 10 words.

We provide here an example of n-gram shuffling.

- Original: The market 's pessimism reflects the gloomy outlook in Detroit
- 1-gram : market pessimism the 's Detroit in The gloomy reflects outlook
- 2-gram : 's pessimism in Detroit The market reflects the gloomy outlook
- 3-gram : The market 's gloomy outlook in pessimism reflects the Detroit
- 4-gram : in Detroit The market 's pessimism reflects the gloomy outlook
- 5-gram : the gloomy outlook in Detroit The market 's pessimism reflects
- 6-gram : outlook in Detroit The market 's pessimism reflects the gloomy
- 7-gram : in Detroit The market 's pessimism reflects the gloomy outlook

Phrase swaps Using constituency trees from the Penn Treebank(Marcus et al., 1994), we define phrases as constituents which don't contain any others within them. (See Fig. 2c or Fig. 3a in the main text.) Phrase swaps thus consist of swapping one phrase with another, and leaving other words intact.

To provide an appropriate control perturbation, we swap two disjoint n-grams, which are the same size as true phrases but cross phrase boundaries.

Adjacent word swaps To better isolate the effect of broken phrase boundaries, we used adjacent word swaps. Adjacent words were chosen randomly, and one swap was performed per sentence.

A.4 Partial linear regression

In order to control for uninteresting explanations of our results, we often make use of a simple method for regressing out confounds. Generally, we want to assess the linear relationship between X and Y, when accounting for the (potentially non-linear) effect of another variable Z. In our experiments, X is always the swap-induced distortion and Y is the swap type, like integer-valued tree distance

or binary-valued in/out phrase. We wish to allow $\mathbb{E}[Y|Z]$ and $\mathbb{E}[X|Z]$ to be any smooth function of Z, which is achieved by the least-squares solution to the following partially linear model:

$$Y \sim \beta_x X + \beta_{\mathbf{z}} \cdot \mathbf{f}(Z)$$

where $\mathbf{f}(z)$ is a vector of several (we use 10) basis functions (we used cubic splines with knots at 10 quantiles) of Z. Both regressions have the same optimal β_x, but the one on the left is computationally simpler (Hansen, 2000). The standard confidence intervals on β_x apply.

Intuitively, the β_x obtained by the partially linear regression above is related to the conditional correlation of X and Y given Z: $\rho(X, Y|Z)$. Like an unconditonal correlation, it will be zero if X and Y are conditionally independent given Z, but not necessarily *vice versa* (both X and Y must be Gaussian for the other direction to be true). To compute conditional rank correlations (which assess a monotonic relationship between X and Y), we rank-transform X and Y (this changes the confidence interval calculations).

We apply this method to swap size in Fig. 2 and attentions in Fig. 5. In these supplemental materials, we will also report the results when X is the binary in/out phrase variable, and Z is PMI. The full p-values and coefficients of the uncontrolled and controlled regressions can be found in Table 1, where we observe that past layer 2, the p-value on phrase boundary is very significant ($p < 10^{-12}$).

A.5 Supervised probes

In this section, we describe the experiments based on the three linguistic tasks: parts of Speech (POS); grandparent tags (GP); and constituency tree distance.

The POS and GP classifiers were multinomial logistic regressions trained to classify each word's POS tag (e.g. 'NNP', 'VB') and the tag of its grandparent in the constituency tree, respectively. If a word has no grandparent, its label is the root token 'S'. The probes were optimized with standard stochastic gradient descent, 50 sentences from the PTB per mini-batch. 10 epochs, at 10^{-3} learning rate, were sufficient to reach convergence.

The distance probe is a linear map $\mathbf{B}$ applied to each word-vector $\mathbf{w}$ in the sentence, and trained such that, for all word pairs i, j, $\text{TreeDist}(i, j)$ matches $\|\mathbf{B}(\mathbf{w}_i - \mathbf{w}_j)\|_2^2$ as closely as possible. Unlike the classifiers, there is freedom in the output dimension of $\mathbf{B}$; we used 100, although performance and results are empirically the same for any choice greater than ~ 64. Our probes are different from (Hewitt and Manning, 2019) in two ways: (1) we use constituency trees, instead of dependency trees, and (2) instead of an L1 loss function, we use the Poisson (negative) log-likelihood as the loss function. That is, if $\lambda_{i,j} = \|\mathbf{B}(\mathbf{w}_i - \mathbf{w}_j)\|_2^2$, and $y_{i,j} = \text{TreeDist}(i, j)$

$$-l_{i,j} = y_{i,j} \log \lambda_{i,j} - \lambda_{i,j} - \log y_{i,j}!$$

Otherwise, the probes are trained exactly as in (Hewitt and Manning, 2019). Specifically, we used standard SGD with 20 sentences from the PTB in each mini-batch, for 40 epochs.

Evaluation A linear model is fit to maximize $p(y|\theta(\mathbf{x}))$, with p a probability function (multinomial for classifiers, Poisson for distance), and $\mathbf{x}$ coming from the unperturbed transformer representation. We evaluate the model on $\tilde{\mathbf{x}}$, which are the representations of the data when generated from a perturbed sentence. We take the average of $\log p(y|\theta(\mathbf{x}_i)) - \log p(y|\theta(\tilde{\mathbf{x}}_i))$ over all the data i in all sentences. For example, all words for the classifiers, and all pairs of words for the distance probe. Concretely, we are just measuring the difference in validation loss of the same probe on the $\mathbf{x}$ data and the $\tilde{\mathbf{x}}$ data. But because the loss is an appropriate probability function, we can interpret the same quantity as a difference in log-likelihood between the distribution conditioned on the regular representation and that conditioned on the perturbed representation. Distortion is similarly computed using the full sentence, providing a number for each swap in each sentence.

	Without PMI		With PMI	
Layer	Coeff.$\times 10^{-2}$	p-value	Coeff. $\times 10^{-2}$	p-value
Emb.	-0.21	5.6×10^{-5}	-0.11	9.4×10^{-2}
1	-0.11	3.4×10^{-2}	-0.05	4.2×10^{-1}
2	-0.74	$< 10^{-16}$	-0.53	2.12×10^{-8}
3	-1.6	$< 10^{-16}$	-1.3	2.2×10^{-16}
4	-2.0	$< 10^{-16}$	-1.4	4.4×10^{-16}
5	-2.1	$< 10^{-16}$	-1.5	8.8×10^{-16}
6	-2.4	$< 10^{-16}$	-1.7	$< 10^{-16}$
7	-2.6	$< 10^{-16}$	-1.7	1.6×10^{-15}
8	-3.4	$< 10^{-16}$	-2.3	$< 10^{-16}$
9	-3.8	$< 10^{-16}$	-2.7	$< 10^{-16}$
10	-4.1	$< 10^{-16}$	-3.0	$< 10^{-16}$
11	-3.8	$< 10^{-16}$	-2.8	$< 10^{-16}$
12	-4.2	$< 10^{-16}$	-3.1	$< 10^{-16}$

Table 1: Coefficients and p-values of the regular ('without PMI') and controlled ('with PMI') regressions of distortion against phrase boundary.