What Happens To BERT Embeddings During Fine-tuning?

Amil Merchant[1] [*] **Elahe Rahimtoroghi**[1] **Ellie Pavlick** [1,2] **Ian Tenney**[1]
[1] Google Research [2] Brown University
{amilmerchant, elahe, epavlick, iftenney}@google.com

Abstract

While there has been much recent work studying how linguistic information is encoded in pre-trained sentence representations, comparatively little is understood about how these models change when adapted to solve downstream tasks. Using a suite of analysis techniques (probing classifiers, Representational Similarity Analysis, and model ablations), we investigate how fine-tuning affects the representations of the BERT model. We find that while fine-tuning necessarily makes significant changes, it does not lead to catastrophic forgetting of linguistic phenomena. We instead find that fine-tuning primarily affects the top layers of BERT, but with noteworthy variation across tasks. In particular, dependency parsing reconfigures most of the model, whereas SQuAD and MNLI appear to involve much shallower processing. Finally, we also find that fine-tuning has a weaker effect on representations of out-of-domain sentences, suggesting room for improvement in model generalization.

1 Introduction

The introduction of unsupervised pre-training for Transformer architectures (Vaswani et al., 2017) has led to significant advances in performance on a range of NLP tasks. Most notably, the popular BERT model (Devlin et al., 2019) topped the GLUE (Wang et al., 2019) leaderboard when it was released, and similar models have continued to improve scores over the past year (Lan et al., 2019; Raffel et al., 2019).

Much recent work has attempted to better understand these models and explain what makes them so powerful. Particularly, behavioral studies (Goldberg, 2019, *inter alia*), diagnostic probing classifiers (Liu et al., 2019, *inter alia*), and unsupervised techniques (Voita et al., 2019a, *inter alia*) have shed light on the representations from the pre-trained models and have shown that they encode a wide variety of linguistic phenomena (Tenney et al., 2019b).

However, in the standard recipe for models such as BERT (Devlin et al., 2019), after a Transformer is initialized with pre-trained weights, it is then trained for a few epochs on a supervised dataset. Considerably less is understood about what happens during this fine-tuning stage. Current understanding is mostly derived from observations about model behavior: fine-tuned Transformers achieve state-of-the-art performance but also can end up learning shallow shortcuts, heuristics, and biases (McCoy et al., 2019b,a; Gururangan et al., 2018; Poliak et al., 2018). Thus, in this work, we seek to understand how the internals of the model–the representation space–change when fine-tuned for downstream tasks. We focus on three widely-used NLP tasks: dependency parsing, natural language inference (MNLI), and reading comprehension (SQuAD), and ask:

- What happens to the encoding of linguistic features such as syntactic and semantic roles? Are these preserved, reinforced, or forgotten as the encoder learns a new task? (Section 4)
- Where in the model are changes made? Are parameter updates concentrated in a small number of layers or are there changes throughout? (Section 5)
- Do these changes generalize or does the newfound behavior only apply to the specific task domain? (Section 6)

We approach these questions with three distinct analysis techniques. Supervised probing classifiers (Tenney et al., 2019b; Hewitt and Man-

[*] Work done as member of the Google AI Residency program https://ai.google/research/join-us/ai-residency/

ning, 2019) give a positive test for specific linguistic phenomena, while Representational Similarity Analysis (RSA; Kriegeskorte et al., 2008) gives a task-agnostic measurement of the change in model activations. Finally, we corroborate the probing and RSA results with two types of model ablations–truncation and partial freezing–and measure their effect on end-task performance.

Taken together, we draw the following conclusions. First, linguistic features are not lost during fine-tuning. Second, fine-tuning tends to affect only the top few layers of BERT, albeit with significant variation across tasks: SQuAD and MNLI have a relatively shallow effect, while dependency parsing involves deeper changes to the encoder. We confirm this by partial-freezing experiments which test how many layers *need* to change to do well on each task and relate this to an estimate of task *difficulty* with layer ablations. Finally, we observe that fine-tuning induces large changes on in-domain examples, but the representations of out-of-domain sentences resemble those of the pre-trained encoder.

2 Related Work

Base model Many recent papers have focused on understanding sentence encoders such as ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019), focusing primarily on the "innate" abilities of the pre-trained ("Base") models. For language models that report perplexity scores, behavioral analyses (Goldberg, 2019; Marvin and Linzen, 2018; Gulordava et al., 2018; Ettinger, 2020) have shown that they capture phenomena like number agreement and anaphora. For Transformer-based models (Vaswani et al., 2017), analyses of attention weights have shown interpretable patterns in their structure (Coenen et al., 2019; Vig and Belinkov, 2019; Voita et al., 2019b; Hoover et al., 2019) and found strong correlations to syntax (Clark et al., 2019). However, other studies have also cast doubt on what conclusions can be drawn from attention patterns (Jain and Wallace, 2019; Serrano and Smith, 2019; Brunner et al., 2019).

More generally, supervised probing models (also known as diagnostic classifiers) make few assumptions beyond the existence of model activations and can test for the presence of a wide variety of phenomena. Adi et al. (2016); Blevins et al. (2018) tested recurrent networks for sentence length, word context, and syntactic properties, while Conneau et al. (2018); Jawahar et al. (2019) use another 10 probing tasks, including sensitivity to bigram shift. Tenney et al. (2019b); Liu et al. (2019); Peters et al. (2018b) introduced task suites that probe for high-level linguistic phenomena such as part-of-speech, entity types, and coreference, while Tenney et al. (2019a) showed that these phenomena are represented in a hierarchical order in the different layers of BERT. Hewitt and Manning (2019) also used a geometrically-motivated probing model to explore syntactic structures, and Wallace et al. (2019) explored the ability of ELMo and BERT to faithfully encode numerical information.[1]

While probing models depend on labelled data, parallel work has studied the same encoders using unsupervised techniques. Voita et al. (2019a) used a form of canonical correlation analysis (PW-CCA; Morcos et al., 2018) to study the layer-wise evolution of representations, while Saphra and Lopez (2019) explored how these representations evolve during training. Abnar et al. (2019) used Representational Similarity Analysis (RSA; Laakso and Cottrell, 2000; Kriegeskorte et al., 2008) to study the effect of context on encoder representations, while Chrupała and Alishahi (2019) correlated them with syntax. Abdou et al. (2019); Gauthier and Levy (2019) also compared these representations to fMRI and eye-tracking data.

Fine-tuning In contrast to the pre-trained models, there have been comparatively few studies on understanding the fine-tuning process. Initial studies of fine-tuned encoders have shown state-of-the-art performance on benchmark suites such as GLUE (Wang et al., 2019) and surprising sample efficiency. However, behavioral studies with challenge sets (McCoy et al., 2019b; Poliak et al., 2018; Ettinger et al., 2018; Kim et al., 2018) have shown limited ability to generalize to out-of-domain data and across syntactic perturbations.

Looking more directly at representations, van Aken et al. (2019) focused on question-answering models with task-specific probing models and clustering analysis. They found evidence of different stages of processing in a fine-tuned BERT model but showed only limited comparisons with the pre-trained encoder. Hao et al. (2019) explored fine-tuning from an optimization perspective, find-

[1] See Belinkov and Glass (2019) and (Rogers et al., 2020) for a survey of probing methods.

ing that pre-training leads to more efficient and stable optimization than random initialization. Peters et al. (2019) also analyzed the effects of fine-tuning with respect to the performance of diagnostic classifiers at various layers. Gauthier and Levy (2019) is closest to our work: while focused on correlating representation to fMRI data, they also studied fine-tuning using RSA and the structural probe of Hewitt and Manning (2019), finding a significant divergence between the final representations of models fine-tuned on different tasks. By comparison, we seek a more general analysis of the internal representations of fine-tuned BERT models and also focus on how they change compared to the pre-trained Base model.

3 Experimental Setup

BERT In this paper, we focus on the effects of fine-tuning for the popular BERT architecture (Devlin et al., 2019). During pre-training, the model is initialized by training on masked language-modeling and next sentence prediction, with an unsupervised corpus from BooksCorpus (Zhu et al., 2015) and English Wikipedia. This model uses WordPiece tokenization (Wu et al., 2016) and prepends input sentences with a `[CLS]` token, used for classification. We use the original TensorFlow (Abadi et al., 2015) implementation of BERT[2] and focus on the 12-layer `bert_base_uncased` variant. We denote the pre-trained model as **Base** and refer to fine-tuned versions by the name of the task.

MNLI A common benchmark for natural language understanding, the MNLI dataset (Williams et al., 2018) contains over 433K sentence pairs annotated with textual entailment information. BERT is adapted to this task by feeding the `[CLS]` token in the last layer through an additional output layer for predictions. We use the parameters and architecture of Devlin et al. (2019) for fine-tuning. Across three trials, the evaluation accuracy of our BERT Base model is 83.3 ± 0.1, slightly lower but comparable to the published score of 84.6.

SQuAD The SQuADv1.1 dataset (Rajpurkar et al., 2016) contains over 100,000 crowd-sourced question-answer pairs, created from a set of Wikipedia articles. The answers are given by contiguous spans from the original question, so the task is integrated into the BERT framework with an additional output layer for predicting the start and end tokens of the answer. We use the parameters and architecture of Devlin et al. (2019) for fine-tuning. Our average F1 score is 89.2 ± 0.2, slightly higher than the published 88.5.

Dependency Parsing We also introduce a BERT model fine-tuned on dependency parsing (Dep). We include this task to present a contrasting perspective from the prior two datasets, in particular since prior research has suggested that much of the information needed to solve dependency parsing is already present in the pre-trained base (Hewitt and Manning, 2019; Goldberg, 2019). Our model is trained on data from the CoNLL 2017 Shared Task (Zeman et al., 2017) and uses the features of BERT as input to a biaffine classifier, similar to Dozat and Manning (2017). The model uses a learning rate of 3×10^{-5} with a 10% warm-up portion, uses an Adam optimizer (Kingma and Ba, 2014), and is trained for 20 epochs. The Labeled Attachment Score (LAS) on the development set for our models is 96.3 ± 0.1.

4 What happens to linguistic features?

Equipped with the models trained on these downstream tasks, we ask how the representation of linguistic features in the fine-tuned models compare to those in the pre-trained model? Recent studies have shown that these robust features do not inform predictions on downstream tasks, with models appearing to use dataset heuristics such as lexical overlap (McCoy et al., 2019b) or word priors (Poliak et al., 2018), but it is an open question whether this is because these features are forgotten entirely or simply are not always used. We approach this with supervised probing models, using two complementary techniques: edge probes (Tenney et al., 2019b) which test for labeling information across several formalisms, and structural probes (Hewitt and Manning, 2019) which measure the representation of syntactic structure.

Notation Let m be the hidden size of the attention layers of a Transformer. Then, we define $\boldsymbol{h}_i^l \in \mathbb{R}^m$ to be the hidden representation of the i-th word at the l-th attention layer. Note that BERT uses subword tokenization, so word representations are aggregated using mean pooling over all subword components.

[2] https://github.com/google-research/bert

		∆ for Baselines		∆ for Fine-tuned Models		
Task	**BERT Base**	**Lexical**	**Randomized**	**MNLI**	**SQuAD**	**Dep**
POS	97.53	-8.99	-13.61	-0.17	**-1.52**	-0.22
Constituents	84.35	-24.13	-12.88	**-2.18**	0.08	**4.35**
Dependencies	95.45	-15.57	-18.19	-0.51	**-2.49**	0.18
Entities	96.19	-6.56	-9.97	-0.34	-0.96	-0.62
SRL	92.87	-13.57	-15.04	-0.36	**-2.90**	-0.50
Coreference	95.72	-5.82	-6.15	-0.49	-0.84	**-1.22**
SPR	84.61	-6.56	-12.21	-0.67	-0.40	**-1.17**
Relations	79.50	-20.68	-40.53	-0.75	-0.37	**-2.53**

Table 1: Comparison of F1 performance on the edge probing tasks before and after fine-tuning. The BERT Base performance is consistent with (Tenney et al., 2019b), and the results show that the fine-tuned models retain most of the linguistic concepts discovered during unsupervised pre-training. We report single numbers for clarity, but note that variation across runs is ±0.5 between probing runs, ±0.7 between fine-tuning runs from the same checkpoint, and ±1.0 point between different pretraining runs.

Edge Probing The edge probing tasks of Tenney et al. (2019b) aim to measure how well a contextual encoder captures linguistic phenomena, ranging from syntactic concepts such as part-of-speech to more semantic abstractions including entity typing and relation classification. To perform this task, the trained encoder is frozen, and the relevant hidden states $\boldsymbol{h}_i^l$ are fed into an auxiliary shallow neural network to predict labeling information. We use the tasks, architecture, and procedure of Tenney et al. (2019b).[3] After training, we report the micro-averaged F1 scores on a held-out test set.

Structural Probe The structural probes of Hewitt and Manning (2019) also analyze the token representations but are designed to evaluate how well this space encodes syntactic structure. Specifically, the probe identifies whether the squared L2 distance of the representations under some linear transformation encodes the tree distances between words in the dependency parse. The initial paper's results showed that the deep models (ELMo and BERT) discover this syntax information during pre-training, which is not present in simple word embedding baselines.

The first structural probe predicts the tree depth (tree distance from the root node) by:

$$||\boldsymbol{h}_i^l||_B = (B\boldsymbol{h}_i^l)^\top (B\boldsymbol{h}_i^l)$$

where $B \in \mathbb{R}^{k \times m}$ is a learned matrix.[4] The corresponding metrics are the prediction accuracy of the root node and the Spearman correlation between the predicted and true tree depths.

The second structural probe measures the pairwise distances for all words in the parse tree. For any two words $(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l)$, the distance is given by:

$$||\boldsymbol{h}_i^l - \boldsymbol{h}_j^l||_B = (B(\boldsymbol{h}_i^l - \boldsymbol{h}_j^l))^\top (B(\boldsymbol{h}_i^l - \boldsymbol{h}_j^l))$$

Following Hewitt and Manning (2019), we evaluate Spearman correlation between each row of the predicted and true distance matricies, and also compute the minimum spanning tree and compare to the true parse using the Undirected Unlabeled Attachment Score (UUAS).

4.1 Results

The results from both probing tasks largely demonstrate that the linguistic structures from pre-training are preserved in the fine-tuned models. This is first seen in the edge probing metrics presented in Table 1. For the sake of comparison, we also provide baseline results on the output of the embedding layer (Lexical) and a randomly initialized BERT architecture (Randomized). These baselines are important because inspection-based analysis can often discover patterns that are not obviously present due to the high capacity of auxiliary classifiers. For example, Zhang and Bowman (2018); Hewitt and Liang (2019) found that expressive-enough probing methods can have decent performance even when trained on random noise.

[3] The dependency labeling task is from the English Web Treebank (Silveira et al., 2014), SPR corresponds to SPR1 from Teichert et al. (2017), and relations is Task 8 from SemEval 2010 (Hendrickx et al., 2010). All of the other tasks are from OntoNotes 5.0 (Weischedel et al., 2013).

[4] For our experiments, we use a rank $k = 512$ to match the projection dimension from the edge probes.

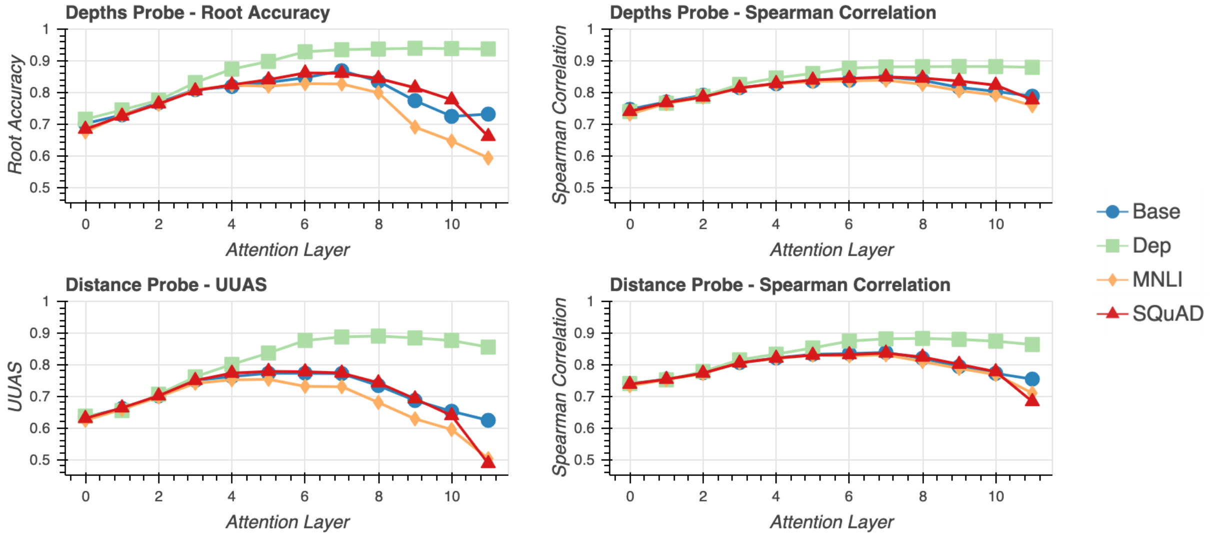

Figure 1: Comparison of the structural probe performance on BERT models before and after fine-tuning. The stability of the Spearman correlations between both the depths and distance probes suggest that the embeddings still retain significant information about the syntax of inputted sentences.

Across the edge probing suite, for all three tasks we see only small changes in F1 score compared to BERT base. In most cases we observe a drop in performance of 0.5-2%, with some variation: MNLI and SQuAD lead to drops of 1.5-3% on syntactic tasks–constituents, and POS, dependencies, and SRL, respectively–while the dependency parsing model leads to significantly improved syntactic performance (+4% on constituent labeling) while dropping performance on the more semantically-oriented coreference, SPR, and relation classification tasks. Nonetheless, in all cases these effects are small: they are comparable to the variation between randomly-seeded fine-tuning runs (±0.7), and much smaller than the difference between the full model and the Lexical or Randomized baselines, suggesting that most linguistic information from BERT is still available after fine-tuning.

Next, we turn to the structural probe, with results seen in Figure 1. First, the dependency parsing fine-tuned model shows improvements in both the correlation and the absolute metrics of root accuracy and UUAS, as early as layer 5. Since the structural probes are designed and trained to look for syntax, this result suggests that the fine-tuning improves the model's internal representation of such information. This makes intuitive sense as the fine-tuning task is aligned with the probing task.

On the MNLI and SQuAD fine-tuned models, we observe small drops in performance, particularly with the final layer. These changes are most pronounced on the root accuracy and UUAS metrics, which score against a discrete decoded solution (`argmin` for root accuracy or minimum spanning tree for UUAS), but are smaller in magnitude on Spearman correlations which consider all predictions. This suggests that while some information is lost, the actual magnitude of change within the "syntactic subspace" is quite small. This is consistent with observations by Gauthier and Levy (2019) and suggests that information about syntactic structure is well-preserved in end-task models.

Overall, the results from these two probing techniques suggest that there is no catastrophic forgetting. This is surprising as a number of prior error analyses have shown that the fine-tuned models often do not use syntax (McCoy et al., 2019b) and rely on annotation artifacts (Gururangan et al., 2018) or simple pattern matching (Jia and Liang, 2017) to solve downstream tasks. Our analysis suggests that the while this linguistic information may not be incorporated into the final predictions, it is still available in the model's representations.

5 What changes in the representations?

Supervised probes are highly targeted: as trained models, they are sensitive to particular linguistic phenomena, but they also can learn to ignore everything else. If the supervised probe is closely related to the fine-tuning task–such as for syntactic probes and a dependency parsing model–we have observed significant changes in performance, but otherwise we see little effect. Nonetheless, we know that *something* must be changing when fine-tuning–as evidenced by prior work that shows that end-task performance degrades if the encoder is completely frozen (Peters et al., 2019). To explore this change more broadly, we turn to an unsupervised technique, Representational Similarity Analysis, and corroborate our findings with layer-based ablations.

5.1 Representational Similarity Analysis

Representational Similarity Analysis (RSA; Laakso and Cottrell, 2000) is a technique for measuring the similarity between two different representation spaces for a given set of stimuli. Originally developed for neuroscience (Kriegeskorte et al., 2008), it has become increasingly used to analyze similarity between neural network activations (Abnar et al., 2019; Chrupała and Alishahi, 2019). The method works by using a common set of n examples, used to create two comparable sets of representations. Using two kernels (possibly different for each representation space) to measure the similarity of paired examples, the sets of representations are then converted into two pairwise similarity matrices in $\mathbb{R}^{n \times n}$. The final similarity score between the two representation spaces is calculated as the Pearson correlation between the flattened upper triangulars of the two similarity matrices.

In our application, we pass ordinary sentences (Wikipedia), sentence-pairs (MNLI), or question-answer pairs (SQuAD) as inputs to the BERT model, and select a random subset ($n = 5000$) of tokens as our stimuli. This choice of input is consistent with the masked language model pre-training and various fine-tuning tasks (such as SQuAD and dependency parsing) in analyzing the contextual representations for every token. We extract representations as the activations of corresponding layers from the two models to compare (e.g. Base vs. a fine-tuned model). Following previous applications of RSA to text representations (Abnar et al., 2019; Chrupała and Alishahi, 2019), we adopt cosine similarity as the kernel for all of our experiments.

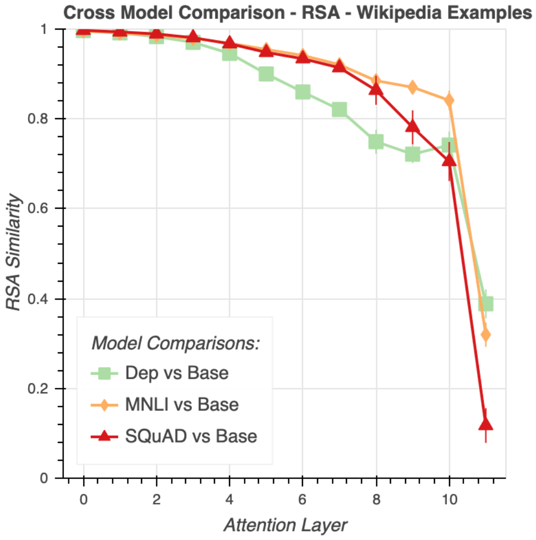

Figure 2: Comparison of the representations from BERT base and the various fine-tuned models, when tested on Wikipedia examples. The dependency probing model starts to diverge from BERT Base around layer 5, which matches the previous results from edge probing. For the MNLI and SQuAD fine-tuned models, the differences from the Base model mainly arise in the top layers of the network.

While RSA does not require learning any parameters and is thus resistant to overfitting (Abdou et al., 2019), the unsupervised technique is sensitive to spurious signals in the representations that may not be relevant to model behavior.[5] To mitigate this, we repeat the BERT pre-training procedure (as described in Section 3 of (Devlin et al., 2019)) from scratch three times. For each pre-trained checkpoints, we fine-tune on the three downstream task and report the average for these independent runs.

Results Figure 2 shows the results of our RSA analysis comparing the three task models, **Dep.**, **MNLI**, and **SQuAD**, to BERT **Base** at each layer, using single-sentence inputs randomly selected from English Wikipedia. Across all tasks, we observe that changes generally arise in the top layers of the network, with very little change observed in the layers closest to the input. To first order, this may be a result of optimization: vanishing gradients result in the most change in the layers closest to the loss. Yet we do observe significant differences between tasks. Except for the output layer which is particularly sensitive to the form of the output (span-based for dependencies and SQuAD, or using the [CLS] token for MNLI), we see that MNLI involves the smallest changes to the model: the second-to-last attention layer still shows a very

[5] We note that probing techniques are more robust to this, since they learn to focus on relevant features.

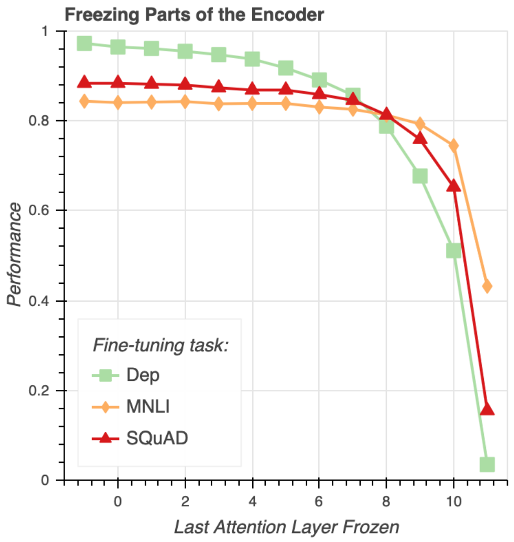

Figure 3: Effects of freezing an increasing number of layers during fine-tuning on performance, where different lines correspond to various tasks (we report the evaluation accuracy for MNLI, F1 score for SQuAD, and LAS for Dep). The point at -1 corresponds to no frozen components. The graph shows that only a few unfrozen layers are needed to improve task performance, supporting the shallow processing conclusion.

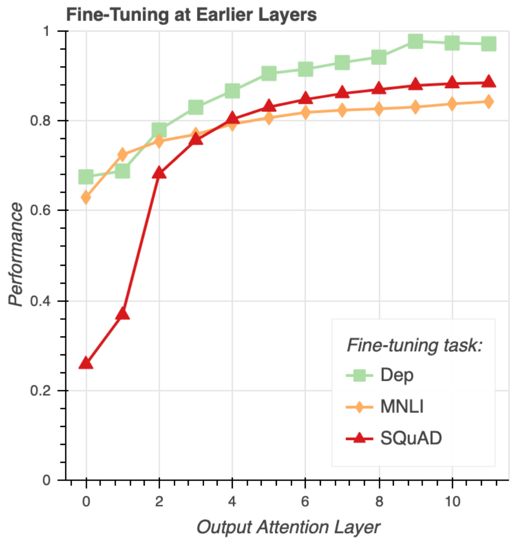

Figure 4: Effects of fine-tuning at earlier layers of BERT. We note that the MNLI evaluation accuracy and SQuAD F1 score approach the full model performance by layer 6, whereas the dependency parsing LAS seems to require more layers. This support our hypothesis that the processing of dataset-specific features is shallow.

high similarity score of 0.84 ± 0.02 compared to the representations of the pretrained encoder. The SQuAD model shows a steeper change, behaving similarly to the Base model through layer 7 but dropping off steeply afterwards - suggesting that fine-tuning on this task involves a deeper, yet still relatively shallow reconfiguration of the encoder.

Finally, dependency parsing presents a dissimilar pattern: we observe the deepest changes, departing from the Base model as early as layers 4 and 5. This is consistent with our supervised probing observations (Section 4), in which we observe improved performance on syntactic features at a similar point in the model (Figure 1).

5.2 Layer Ablations

As an unsupervised, metric-based technique, RSA tells us about broad changes in the representation space, but does not in itself say if these changes are important for the model's behavior–i.e. for the processing necessary to solve the downstream task. To measure our observations in terms of task performance, we turn to two layer ablation studies.

Partial Freezing can be thought of as a test for how many layers *need* to change for a downstream task. We freeze the bottom k layers (and the embeddings)–treating them as features–but allow the rest to adapt. Effectively, this clamps the first k layers to have RSA similarity of 1 with the Base model. Also, we perform **model truncation** as a rough estimate of difficulty for each task, and as an attempt to de-couple the results of partial freezing from helpful features that may be available in top layers of BERT Base (Tenney et al., 2019a). Figure 3 (partial freezing) Figure 4 (truncation) show the effect on task performance.

The patterns we observe corroborate the findings of our RSA analysis. On MNLI, we find that performance does not drop significantly unless the last two layers are frozen, while the truncated models are able to achieve comparable performance with only three attention layers. This suggests that while natural language inference (Dagan et al., 2006) is known to be a complex task *in the limit*, most MNLI examples can be resolved with relatively shallow processing. SQuAD exhibits a similar trend: we see a significant performance drop when 3 or fewer layers are allowed to change (e.g. freezing through layer 8 or higher), consistent with where RSA finds the greatest change. From our truncation experiment, we similarly see that only five layers are needed to achieve comparable performance to the full model.

Dependency parsing performance drops more rapidly still–in both experiments–again consistent with the previous results from RSA. This is per-

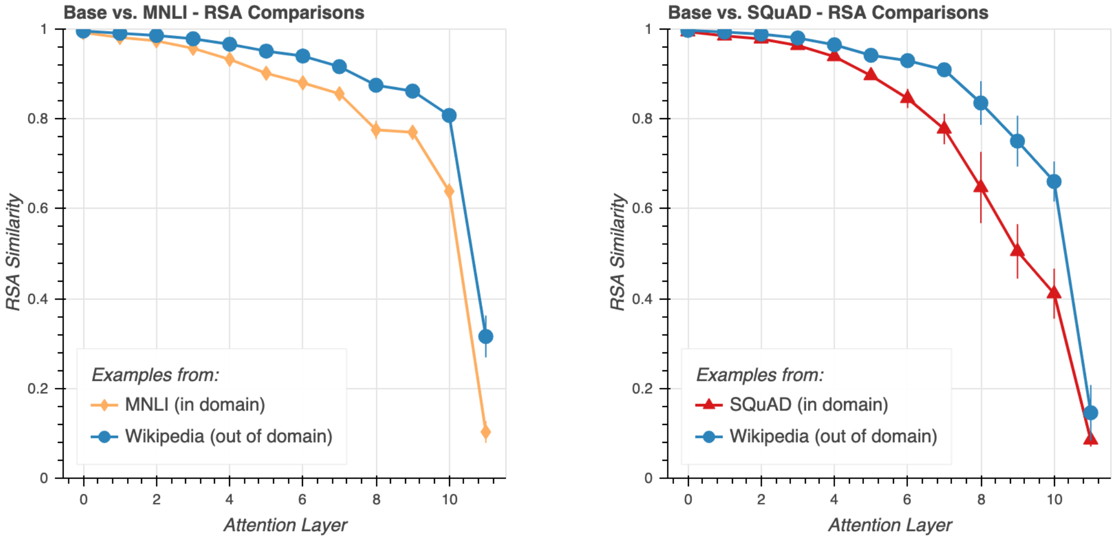

Figure 5: Comparison of the representations in the MNLI (left) and SQuAD (right) fine-tuned models and those of BERT Base, with the different lines corresponding to examples coming from various datasets. These graphs show that fine-tuning models only lead to shallow changes, consolidated to the last few layers. Also, we see that fine-tuning has a much greater impact on the token representations of in-domain data.

haps surprising, since probing analysis (Goldberg, 2019; Marvin and Linzen, 2018) suggests that many syntactic phenomena are well-captured by the pre-trained model, and diagnostics for dependency parsing in particular (Tenney et al., 2019b,a; Hewitt and Manning, 2019; Clark et al., 2019) show strong performance from probes on frozen models. Yet as observed with the structural probes (Figure 1) there is much headroom available, and it appears that to capture it requires changing deeper parts of the model. We hypothesize that this effect may come from the hierarchical nature of parsing, which requires additional layers to determine the tree-like structure, and fully reconciling these observations would be a promising direction for future work.

6 Out-of-Domain Behavior

Finally, we ask whether the effects of fine-tuning are general: do they apply only to in-domain inputs, or do they lead to broader changes in behavior? This is usually explored by behavioral methods, in which a model is trained on one domain and evaluated on another–for example, the mismatched evaluation for MNLI (Williams et al., 2018)–but this analysis is limited by the availability of labeled data. By using Representational Similarity Analysis, we can test this in an unsupervised manner.

We use RSA to compare the fine-tuned model to Base and observe the degree of similarity when inputs are drawn from different corpora. We use random samples from the development sets for MNLI (as `premise [SEP] hypothesis`) and SQuAD (as `question [SEP] passage`) as in-domain for their respective models,[6] and as the out-of-domain control we use random Wikipedia sentences (which resemble the pre-training domain). As in Section 5.1, we use the representations of $n = 5000$ tokens as our stimuli for each comparison.[7] Results for the MNLI and SQuAD fine-tuned models are shown in Figure 5.

We note that in both cases, the trend follows Section 5.1 in that the models diverge from BERT Base in the top layers, but there is a significantly larger change in representations on the in-domain examples. When evaluated on Wikipedia sentences, which resemble the pre-training data, the similarity scores are much higher. This suggests that fine-tuning leads the model to change its representations for the fine-tuning domain but to continue to behave more like the Base model otherwise.

[6] Note that these are unseen during fine-tuning, although RSA scores do not change significantly if the MNLI or SQuAD training sets are used.

[7] We also tested single-sentence examples from MNLI and SQuAD by only taking the premise and question respectively; the trends were similar to Figure 5.

7 Conclusions

In this paper, we employ three complementary analysis methods to gain insight on the effects of fine-tuning on the representations produced by BERT. From supervised probing analyses, we find that the linguistic structures discovered during pre-training remain available after fine-tuning. Prior studies (McCoy et al., 2019b; Jia and Liang, 2017) have shown that end-task models often fall back on simple heuristics; taken together, our results suggest that while present, perhaps the linguistic features are not always incorporated into final predictions.

Next, our results using RSA and layer ablations show that the changes from fine-tuning alter a fraction of the model capacity, specifically within the top few layers (up to some variation across tasks). Also, although fine-tuning has a significant affect on the representations of in-domain sentences, the representations of out-of-domain examples remain much closer to those of the pre-trained model.

Overall, these conclusions suggest that fine-tuning–as currently practiced–is a conservative process: largely preserving linguistic features, affecting only a few layers, and specific to in-domain examples. While the standard fine-tuning recipe undeniably leads to strong performance on many tasks, there appears to be room for improvement: an opportunity to refine this transfer step–potentially by utilizing more of the model capacity–to better the generalization and transferability.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Mostafa Abdou, Artur Kulmizev, Felix Hill, Daniel M. Low, and Anders Søgaard. 2019. Higher-order comparisons of sentence encoder representations. 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 5837–5844, Hong Kong, China. Association for Computational Linguistics.

Samira Abnar, Lisa Beinborn, Rochelle Choenni, and Willem Zuidema. 2019. Blackbox meets blackbox: Representational similarity & stability analysis of neural language models and brains. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 191–203, Florence, Italy. Association for Computational Linguistics.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207.

Betty van Aken, Benjamin Winter, Alexander Lser, and Felix A. Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations.

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

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Melbourne, Australia. Association for Computational Linguistics.

Gino Brunner, Yang Liu, Damin Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2019. On identifiability in transformers.

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.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of BERT. *CoRR*, abs/1906.02715.

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.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg. Springer-Verlag.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR (Poster)*. OpenReview.net.

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.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jon Gauthier and Roger Levy. 2019. Linking artificial and human neural representations of language. 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 529–539, Hong Kong, China. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. 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 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2019. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*.

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.

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

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Juho Kim, Christopher Malon, and Asim Kadav. 2018. Teaching syntax by adversarial distraction. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

N. Kriegeskorte, M. Mur, and P. Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Front Syst Neurosci*, 2:4.

Aarre Laakso and Garrison Cottrell. 2000. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76.

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. 2019. 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.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

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.

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.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

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.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew R Gormley. 2017. Semantic proto-role labeling. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

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.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in 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 5306–5314, Hong Kong, China. 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*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağr Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran

Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Kelly W Zhang and Samuel R Bowman. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. *arXiv preprint arXiv:1809.10040*.

Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.