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 much recent work has examined 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—supervised probing, unsupervised similarity analysis, and layer-based ablations—we investigate how fine-tuning affects the representations of the BERT model. We find that while fine-tuning necessarily makes some significant changes, there is no catastrophic forgetting of linguistic phenomena. We instead find that fine-tuning is a conservative process that primarily affects the top layers of BERT, albeit with noteworthy variation across tasks. In particular, dependency parsing reconfigures most of the model, whereas SQuAD and MNLI 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

Unsupervised pre-training of deep language models has led to significant advances on many NLP tasks, with the popular BERT model (Devlin et al., 2019) and successors (e.g. Lan et al., 2019; Raffel et al., 2020) dominating the GLUE leaderboard (Wang et al., 2019) and other benchmarks over the past year. Many recent works have attempted to better understand these models and explain what makes them so powerful. Particularly, behavioral studies (e.g. Marvin and Linzen, 2018; Goldberg, 2019), diagnostic probing classifiers (e.g. Veldhoen et al., 2016; Belinkov et al., 2017; Hupkes et al., 2018), and unsupervised techniques (e.g. Saphra and Lopez, 2019; Voita et al., 2019a) 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; Liu et al., 2019).

However, in the standard recipe for models such as BERT (Devlin et al., 2019), after initializing with pre-trained weights, they are 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 based largely on the models' performance. While fine-tuned Transformers achieve state-of-the-art accuracy, they also can end up learning shallow heuristics (McCoy et al., 2019b; Gururangan et al., 2018; Poliak et al., 2018), suggesting a disconnect between the richness of features learned from pre-training and those used by fine-tuned models. 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? Do different tasks change how shallowly this information is encoded? (Section 4)
- Where in the model are changes made? Are parameter updates concentrated in a small number of layers or are there changes throughout the model? (Section 5)
- Do these changes generalize or does the new behavior only apply to the specific domain on which fine-tuning occurred? (Section 6)

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

Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 33–44

We approach these questions with three complementary analysis techniques. Supervised probing classifiers (Tenney et al., 2019b; Hewitt and Manning, 2019; Voita and Titov, 2020) provide a means of explicitly testing for the presence of pre-specified 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 our results with two types of layer-based ablations–truncation and partial freezing–and measure their effect on end-task performance.

Taken together, we conclude that fine-tuning involves primarily shallow model changes, evidenced by three specific observations. First, linguistic features are not lost during fine-tuning but tasks can differ in how they either surface or obfuscate different phenomena. Second, fine-tuning tends to affect only the top few layers of BERT, albeit with 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 respect to the pre-training regime) via layer ablations. Finally, we observe that fine-tuning induces large changes on in-domain examples, yet on out-of-domain sentences, the representations more closely resemble those of the pre-trained model.

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 example, analyses of attention weights have shown interpretable patterns (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). Kovaleva et al. (2019) also saw that fine-tuning mainly changes the attention of the last few layers, consistent with our findings in Section 5.1. However, other studies have 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 and diagnostic classifiers make few assumptions beyond the existence of model activations and can test for the presence of a wide variety of phenomena. 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 within the layers of BERT. Hewitt and Manning (2019) used a geometrically-motivated probe to explore syntactic structures, and Voita and Titov (2020) and Pimentel et al. (2020) designed information-theoretic techniques that can measure the model and data complexity.[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 (PWCCA; 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.

Fine-tuning Comparatively few analyses have focused 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 (Peters et al., 2018a). 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. van Aken et al. (2019) focused on question-answering models with task-specific probes. Peters et al. (2019) analyzed the effects of fine-tuning with respect to the performance of diagnostic classifiers. Gauthier and Levy (2019) studied fine-tuning via RSA, finding a significant divergence between the representations of models fine-tuned on different tasks. Concurrent work by Tamkin et al. (2020) investigated the transferability of pre-trained language models and performed an number of layer ablations. Consistent with our observations in Section 5.2, they find

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

differences in which layers are important for fine-tuning different tasks. However, none of the prior provides a comprehensive analysis of what happens to the internal representations of the BERT model. In our work, we find that by comparing the Base to the fine-tuned models either via probing, RSA, and layer ablations provides novel insights about this additional phase of training.

3 Experimental Setup

BERT We focus on the popular BERT model (Devlin et al., 2019), focusing on the 12-layer `base_uncased` variant.[2] 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. We fine-tune BERT using the architecture and parameters of Devlin et al. (2019), using a softmax layer on `[CLS]` representation to predict the output label. 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 100K crowd-sourced question-answer pairs, created from a set of Wikipedia articles. We fine-tune BERT using the architecture and parameters of Devlin et al. (2019), which uses two independent softmax layers to predict the start and end tokens of the answer span. 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, since prior research has suggested that much of the information needed to solve dependency parsing is already present after pre-training (Hewitt and Manning, 2019; Goldberg, 2019; Tenney et al., 2019b). 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 is 96.3 ± 0.1 for our model[3]

4 What happens to linguistic features?

Equipped with the models trained on these downstream tasks, we ask how the representation of linguistic features compare to those in the pre-trained model? Recent studies have shown that these robust features are not necessarily used to 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 explore this with supervised probing techniques, using edge probing (Tenney et al., 2019b) and structural probes (Hewitt and Manning, 2019) to explore how well linguistic information can be recovered from the fine-tuned model.

Edge Probing Edge probing aims to measure how contextual representations encode various linguistic phenomena, including part-of-speech, entity typing, and coreference. We use the tasks and parameters of Tenney et al. (2019b), which uses a two-layer MLP to predict edge and span labels from frozen encoder representations.[4] As we are interested in whether the linguistic knowledge is retained by the model overall, we utilize the *mix* version of the edge probes, which takes as input a learned scalar mixing of the representations from every layer.[5] After training, we report the micro-averaged F1 scores on a held-out test set.

Structural Probe Complementary to the edge probes, the structural probes of Hewitt and Manning (2019) analyze how well representations encode syntactic structure. Specifically, the probe identifies whether the squared L2 distance of representations under some linear transformation en-

[2] We use the original TensorFlow (Abadi et al., 2015) implementation from `https://github.com/google-research/bert`.

[3] We provide additional details of the experiments and datasets in Appendix C for the purpose of reproducibility.

[4] 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).

[5] We also explored the effects of fine-tuning on the *top* layer of BERT to provide additional insight into whether this linguistic information may be lost from the top layers even if still present elsewhere. For results, see Appendix A.

		Δ for Baselines		Δ for Fine-tuned Models		
Task	**BERT Base**	**Lexical**	**Randomized**	**MNLI**	**SQuAD**	**Dep**
POS	97.5	-9.0	-13.6	-0.2	**-1.5**	-0.2
Constituents	84.4	-12.9	-24.1	**-2.2**	0.1	**4.4**
Dependencies	95.5	-15.6	-18.2	-0.5	**-2.5**	0.2
Entities	96.2	-6.6	-10.0	-0.3	-0.9	-0.6
SRL	92.9	-13.6	-15.0	-0.4	**-2.9**	-0.5
Coreference	95.7	-5.8	-6.2	-0.5	-0.8	**-1.2**
SPR	84.6	-6.6	-12.2	-0.7	-0.4	**-1.2**
Relations	79.5	-20.7	-40.5	-0.8	-0.4	**-2.5**

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 pre-training runs.

codes the dependency parse. The two versions of the structural probe either attempt to predict the tree depth for each word (distance from the root node) or pairwise distances for all words in the parse tree. For both, we measure the Spearman correlation between predicted and true values [6]

4.1 Results

The results from both probing tasks demonstrate that the linguistic features 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 provide baseline results on the output of the embedding layer (Lexical) and a randomly initialized BERT architecture (Randomized). These baselines are important as 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 perform surprisingly well even when trained on randomized encoders.

Across the edge probing suite, we see only small changes in F1 score from the fine-tuned models 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. We hypothesize that these changes relate to the similarity between tasks: a task like constituent labels help improve dependency parsing, and is thus strengthened, whereas higher level semantic tasks such as SPR contribute less directly and such information may be lost during fine-tuning. Nonetheless, in most 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 within the model 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 the Spearman correlation, 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 minimal changes in performance, with small drops within the final layer. This artifact likely emerges from the fine-tuning setup where the last layer is only needed for classification or span prediction and therefore is unlikely to also retain all the linguistic information.[7]

[6] Note that Hall Maudslay et al. (2020) has recently raised concern about these metrics, but we follow the original method of Hewitt and Manning (2019) for the most comparable results.

[7] A similar story emerges when repeating the edge probing models on the last layer of BERT; see Appendix A.

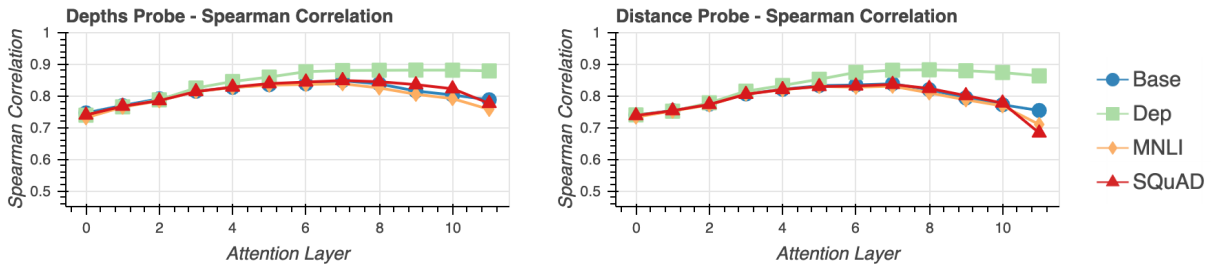

Figure 1: Comparison of the structural probe performance on BERT models before and after fine-tuning. The stability of the Spearman correlations between both the depths and distance probes suggest that the embeddings still retain significant information about the syntax of inputted sentences.

This result suggests that 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 models on downstream tasks.

One caveat of the experimentation above is that it uses complex diagnostic classifiers and only reports final model performance. Instead, what if the linguistic features were simply becoming more difficult to extract from the representations? Then, they could be not as readily "available" after fine-tuning. We explored this hypothesis using Minimum Description Length probes (Voita and Titov, 2020), with the results presented in Appendix B. We found minimal differences across most tasks, where the only significant result was that fine-tuning on dependency parsing made the corresponding edge probing task easier to learn as a function of the number of examples.

4.2 Conclusion

Overall, our results suggest that linguistic features are still available, and that the fine-tuning process does not lead to catastrophic forgetting. Nonetheless, behavioral analyses have shown that fine-tuned models can still fail to leverage even simple syntactic knowledge in their predictions (McCoy et al., 2019b,a; Min et al., 2020), and may instead rely on annotation artifacts (Gururangan et al., 2018) or pattern matching (Jia and Liang, 2017). This suggests that the changes from fine-tuning are conservative: rich features are still present even if the model ends up finding a naive, simple solution.

5 Where do the representations change?

The supervised probes from the previous section 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 dependency parsing–we observe significant changes in performance, but otherwise we see little effect. Nonetheless, we know that *something* must be changing during fine-tuning–at minimum because, as shown in Peters et al. (2019), performance degrades significantly if the encoder is completely frozen. To explore this change, we turn to an unsupervised technique, Representational Similarity Analysis (RSA; Laakso and Cottrell, 2000), which is sensitive to the global structure of the embedding space, and corroborate our findings with layer-based ablations. While these techniques are not targeted to specific linguistic phenomena, they do provide a powerful exploratory tool that can illuminate which parts of the model change and how they vary across datasets.

5.1 Representational Similarity Analysis

RSA 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 sets of representations. For each set, a kernel is used to define a pairwise similarity matrix 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 sample ($n = 5000$) of

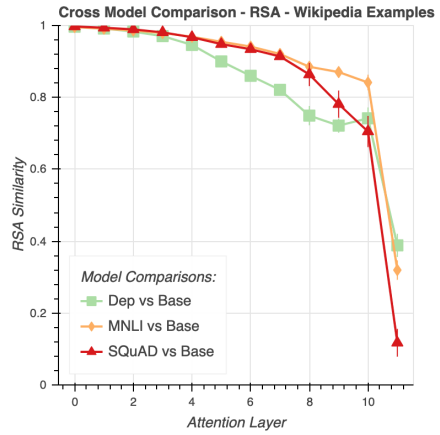

Figure 2: Comparison of the representations from BERT base and various fine-tuned models, when tested on Wikipedia examples. The dependency probing model starts to diverge from BERT Base around layer 5, matching previous results from edge probing. For the MNLI and SQuAD models, the differences from the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

tokens as stimuli. This input is consistent with the masked language model pre-training, various fine-tuning tasks, and diagnostic classifiers in analyzing the contextual representations for every token. We extract 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 the cosine similarity kernel.

While RSA does not require learning any parameters and is thus resistant to overfitting (Abdou et al., 2019), the metric can be sensitive to spurious signals in the representations that may not be relevant to model behavior.[8] 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. Note that in these figures, lower values imply greater change relative to the pre-trained model. Across all tasks, we observe that changes generally arise in the top layers of the network, with 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 interestingly do observe significant differences between tasks. For dependency parsing, we observe the deepest changes, departing from the Base model as early as layers 4 and 5. This result likely arises as syntactic understanding of input is maximized in the early layers of the model, as measured by the edge probes of (Tenney et al., 2019a) and presented structural probes. Performing optimally on this task would require surfacing this information in all subsequent layers, leading to these changes.

Except for the last 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 high similarity score of 0.84 ± 0.02 compared to the representations of the pre-trained encoder. The SQuAD model shows a slightly steeper change, behaving similarly to the Base model through layer 7 but dropping off afterwards - suggesting that fine-tuning on this task involves a deeper, yet still relatively shallow reconfiguration of the encoder. SQuAD likely shows deeper processing as choosing an answer span still requires satisfying a number of syntactic constraints and requires evolution across more than just two layers (van Aken et al., 2019), but overall, we see that for these benchmark tasks, fine-tuning is conservative and only changes a fraction of the model's representations.

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

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

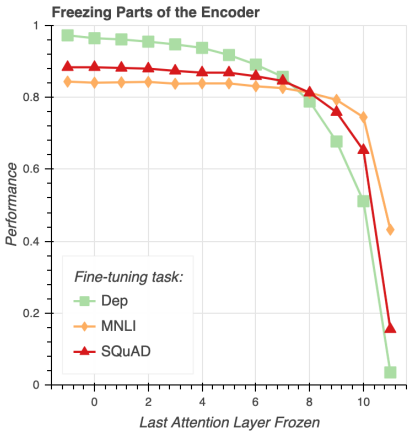

Figure 3: Effects of freezing an increasing number of layers during fine-tuning on performance (we report the evaluation accuracy for MNLI, F1 score for SQuAD, and LAS for Dep). The point at -1 corresponds to no frozen components. The graph shows that only a few unfrozen layers are needed to improve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

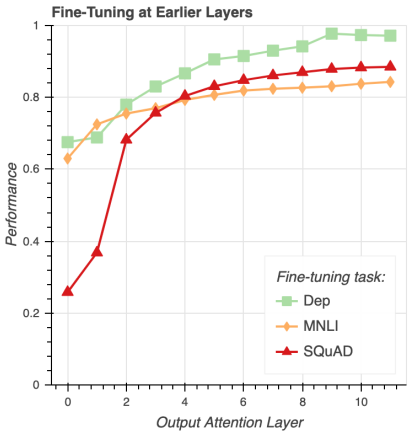

Figure 4: Effects of fine-tuning at earlier layers of BERT. We note that the MNLI evaluation accuracy and SQuAD F1 score approach the full model performance by layer 6, whereas the dependency parsing LAS seems to require more layers.

freezing from helpful features that may be available in top layers of BERT Base (Tenney et al., 2019a). Figure 3 (partial freezing) and 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 even more rapidly–in both experiments–consistent with the results from RSA. This is 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 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 full tree structure. 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 inputs that look like the fine-tuning data, 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 RSA, 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,[9] 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

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

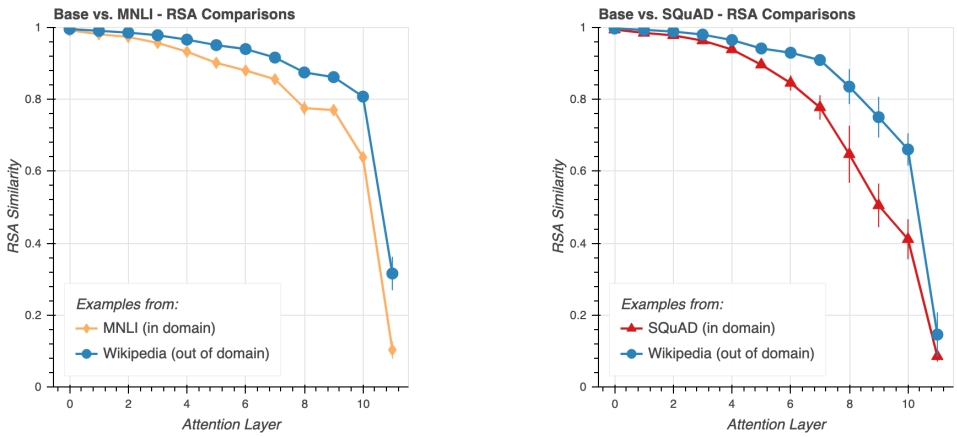

Figure 5: Comparison of the representations in the MNLI (left) and SQuAD (right) fine-tuned models and those of BERT Base, with the different lines corresponding to examples coming from various datasets. These graphs show that fine-tuning models only lead to shallow changes, consolidated to the last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

representations of $n = 5000$ tokens as our stimuli for each comparison.[10] Results for the MNLI and SQuAD fine-tuned models are shown in Figure 5.

Although we see that all models diverge from BERT Base in the top layers, there is a significantly larger change in the representations on in-domain examples. This suggests that fine-tuning is specific to the target domain. For other examples, such as the Wikipedia sentences which resemble the pre-training data, the similarity score with BERT Base is much higher. This suggests that fine-tuning leads the model to change its representations for the new domain but to continue to behave more like the Base model otherwise. This final result again shows that fine-tuning is conservative and suggests room for improvement in model generalization to out-of-domain sentences.

7 Conclusions

In this paper, we employ three complementary analysis methods to gain insight into 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, though this information is not strengthened by tuning on benchmark tasks such as MNLI and SQuAD. In light of prior studies (McCoy et al., 2019b; Jia and Liang, 2017) which have shown that end-task models often fall back on simple heuristics, our results are especially interesting: they suggest that the model has the option of using stronger features, but chooses to use heuristics instead.

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 effect 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: 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.

Finally, in this work, we pulled from a range of analysis techniques to understand very fine-grained aspects of model representations (via probing classifiers) and coarse-grained ones (via RSA). An important direction for future work is the development of new techniques which allow for more exploration of the middle ground. Given available techniques, we can illuminate broadly that models are changing and test hypotheses about specific features (with probing tasks or attention analyses). New principled methods for discovering which features change will be invaluable for a deeper understanding of these models.

[10] 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.

Acknowldgements

We thank our anonymous reviewers for their helpful feedback; Deepak Ramachandran, Kelvin Guu, and Slav Petrov for providing feedback on an early draft of this paper; and Tim Dozat for his help implementing the fine-tuning task for dependency parsing.

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.

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, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

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

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.

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, 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.

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.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. 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*.

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

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.

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.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. *arXiv preprint arXiv:2004.11999*.

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.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*.

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.

Yada Pruksachatkun, Phil Yeres, Haokun Liu, Jason Phang, Phu Mon Htut, Alex Wang, Ian Tenney, and Samuel R Bowman. 2020. jiant: A software toolkit for research on general-purpose text understanding models. *arXiv preprint arXiv:2003.02249*.

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

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.

J. Rissanen. 1984. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4):629–636.

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)*.

Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. 2020. Investigating transferability in pretrained language models. *arXiv preprint arXiv:2004.14975*.

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*.

Sara Veldhoen, Dieuwke Hupkes, Willem H Zuidema, et al. 2016. Diagnostic classifiers revealing how neural networks process hierarchical structure. In *CoCo@ NIPS*, pages 69–77.

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.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*.

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.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

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 Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361.

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.