On the Interplay Between Fine-tuning and Sentence-level Probing for Linguistic Knowledge in Pre-trained Transformers

Marius Mosbach **Anna Khokhlova** **Michael A. Hedderich** **Dietrich Klakow**
Spoken Language Systems (LSV)
Department of Language Science and Technology
Saarland Informatics Campus, Saarland University, Germany
{mmosbach,akhokhlova,mhedderich,dklakow}@lsv.uni-saarland.de

Abstract

Fine-tuning pre-trained contextualized embedding models has become an integral part of the NLP pipeline. At the same time, probing has emerged as a way to investigate the linguistic knowledge captured by pre-trained models. Very little is, however, understood about how fine-tuning affects the representations of pre-trained models and thereby the linguistic knowledge they encode. This paper contributes towards closing this gap. We study three different pre-trained models: BERT, RoBERTa, and ALBERT, and investigate through sentence-level probing how fine-tuning affects their representations. We find that for some probing tasks fine-tuning leads to substantial changes in accuracy, possibly suggesting that fine-tuning introduces or even removes linguistic knowledge from a pre-trained model. These changes, however, vary greatly across different models, fine-tuning and probing tasks. Our analysis reveals that while fine-tuning indeed changes the representations of a pre-trained model and these changes are typically larger for higher layers, only in very few cases, fine-tuning has a positive effect on probing accuracy that is larger than just using the pre-trained model with a strong pooling method. Based on our findings, we argue that both positive and negative effects of fine-tuning on probing require a careful interpretation.

1 Introduction

Transformer-based contextual embeddings like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b) and ALBERT (Lan et al., 2020) recently became the state-of-the-art on a variety of NLP downstream tasks. These models are pre-trained on large amounts of text and subsequently fine-tuned on task-specific, supervised downstream tasks. Their strong empirical performance triggered questions concerning the linguistic knowledge they encode in their representations and how it is affected by the training objective and model architecture (Kim et al., 2019; Wang et al., 2019a). One prominent technique to gain insights about the linguistic knowledge encoded in pre-trained models is *probing* (Rogers et al., 2020). However, works on probing have so far focused mostly on pre-trained models. It is still unclear how the representations of a pre-trained model change when fine-tuning on a downstream task. Further, little is known about whether and to what extent this process adds or removes linguistic knowledge from a pre-trained model. Addressing these issues, we are investigating the following questions:

1. How and where does fine-tuning affect the representations of a pre-trained model?

2. To which extent (if at all) can changes in probing accuracy be attributed to a change in linguistic knowledge encoded by the model?

To answer these questions, we investigate three different pre-trained encoder models, BERT, RoBERTa, and ALBERT. We fine-tune them on sentence-level classification tasks from the GLUE benchmark (Wang et al., 2019b) and evaluate the linguistic knowledge they encode leveraging three sentence-level probing tasks from the SentEval probing suite (Conneau et al., 2018). We focus on sentence-level probing tasks to measure linguistic knowledge encoded by a model for two reasons: 1) during fine-tuning we explicitly train a model to represent sentence-level context in its representations and 2) we are interested in the extent to which this affects existing sentence-level linguistic knowledge already present in a pre-trained model.

We find that while, indeed, fine-tuning affects a model's sentence-level probing accuracy and these effects are typically larger for higher layers, changes in probing accuracy vary depend-

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2502–2516
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

ing on the encoder model, fine-tuning and probing task combination. Our results also show that sentence-level probing accuracy is highly dependent on the pooling method being used. **Only in very few cases, fine-tuning has a positive effect on probing accuracy that is larger than just using the pre-trained model with a strong pooling method.** Our findings suggest that changes in probing performance can not exclusively be attributed to an improved or deteriorated encoding of linguistic knowledge and should be carefully interpreted. We present further evidence for this interpretation by investigating changes in the attention distribution and language modeling capabilities of fine-tuned models which constitute alternative explanations for changes in probing accuracy.

2 Related Work

Probing A large body of previous work focuses on analyses of the internal representations of neural models and the linguistic knowledge they encode (Shi et al., 2016; Ettinger et al., 2016; Adi et al., 2016; Belinkov et al., 2017; Hupkes et al., 2018). In a similar spirit to these first works on probing, Conneau et al. (2018) were the first to compare different sentence embedding methods for the linguistic knowledge they encode. Krasnowska-Kieraś and Wróblewska (2019) extended this approach to study sentence-level probing tasks on English and Polish sentences.

Alongside sentence-level probing, many recent works (Peters et al., 2018; Liu et al., 2019a; Tenney et al., 2019b; Lin et al., 2019; Hewitt and Manning, 2019) have focused on token-level probing tasks investigating more recent contextualized embedding models such as ELMo (Peters et al., 2018), GPT (Radford et al., 2019), and BERT (Devlin et al., 2019). Two of the most prominent works following this methodology are Liu et al. (2019a) and Tenney et al. (2019b). While Liu et al. (2019a) use linear probing classifiers as we do, Tenney et al. (2019b) use more expressive, non-linear classifiers. However, in contrast to our work, most studies that investigate pre-trained contextualized embedding models focus on pre-trained models and not fine-tuned ones. Moreover, we aim to assess how probing performance changes with fine-tuning and how these changes differ based on the model architecture, as well as probing and fine-tuning task combination.

Fine-tuning While fine-tuning pre-trained language models leads to a strong empirical performance across various supervised NLP downstream tasks (Wang et al., 2019b), fine-tuning itself (Dodge et al., 2020) and its effects on the representations learned by a pre-trained model are poorly understood. As an example, Phang et al. (2018) show that downstream accuracy can benefit from an intermediate fine-tuning task, but leave the investigation of why certain tasks benefit from intermediate task training to future work. Recently, Pruksachatkun et al. (2020) extended this approach using eleven diverse intermediate fine-tuning tasks. They view probing task performance after fine-tuning as an indicator of the acquisition of a particular language skill during intermediate task fine-tuning. This is similar to our work in the sense that probing accuracy is used to understand how fine-tuning affects a pre-trained model. Talmor et al. (2019) try to understand whether the performance on downstream tasks should be attributed to the pre-trained representations or rather the fine-tuning process itself. They fine-tune BERT and RoBERTa on a large set of symbolic reasoning tasks and find that while RoBERTa generally outperforms BERT in its reasoning abilities, the performance of both models is highly context dependent.

Most similar to our work is the contemporaneous work by Merchant et al. (2020). They investigate how fine-tuning leads to changes in the representations of a pre-trained model. In contrast to our work, their focus, however, lies on edge-probing (Tenney et al., 2019b) and structural probing tasks (Hewitt and Manning, 2019) and they study only a single pre-trained encoder: BERT. We consider our work complementary to them since we study sentence-level probing tasks, use different analysis methods and investigate the impact of fine-tuning on three different pre-trained encoders: BERT, RoBERTa, and ALBERT.

3 Methodology and Setup

The focus of our work is on studying how fine-tuning affects the representations learned by a pre-trained model. We assess this change through sentence-level probing tasks. We focus on sentence-level probing tasks since during fine-tuning we explicitly train a model to represent sentence-level context in the CLS token.

The fine-tuning and probing tasks we study concern different linguistic levels, requiring a model

Model	Task			
	CoLA	SST-2	RTE	SQuAD
Devlin et al. (2019)	52.1	93.5	66.4	80.8/88.5
BERT	59.5	92.4	64.6	78.6/86.5
RoBERTa	60.3	93.6	73.6	81.7/89.3
ALBERT	45.8	88.5	69.6	79.9/87.6

Table 1: Fine-tuning performance on the development set on selected down-stream tasks. For comparison we also report the fine-tuning accuracy of BERT-base-cased as reported by Devlin et al. (2019) on the test set of each of the tasks taken from the GLUE and SQuAD leaderboards. We report Matthews correlation coefficient for CoLA, accuracy for SST-2 and RTE, and exact match (EM) and F_1 score for SQuAD.

to focus more on syntactic, semantic or discourse information. The extent to which knowledge of a particular linguistic level is needed to perform well differs from task to task. For instance, to judge if the syntactic structure of a sentence is intact, no deep discourse understanding is needed. Our hypothesis is that if a pre-trained model encodes certain linguistic knowledge, this acquired knowledge should lead to a good performance on a probing task testing for the same linguistic phenomenon. Extending this hypothesis to fine-tuning, one might argue that if fine-tuning introduces new or removes existing linguistic knowledge into/from a model, this should be reflected by an increase or decrease in probing performance.[1] However, we argue that **encoding or forgetting linguistic knowledge is not necessarily the only explanation for observed changes in probing accuracy**. Hence, the goal of our work is to test the above-stated hypotheses assessing the interaction between fine-tuning and probing tasks across three different encoder models.

3.1 Fine-tuning tasks

We study three fine-tuning tasks taken from the GLUE benchmark (Wang et al., 2019b). All the tasks are sentence-level classification tasks and cover different levels of linguistic phenomena. Additionally, we study models fine-tuned on SQuAD (Rajpurkar et al., 2016) a widely used question answering dataset. Statistics for each of the tasks can be found in the Appendix.

CoLA The Corpus of Linguistic Acceptability (Warstadt et al., 2018) is an acceptability task which tests a model's knowledge of grammatical concepts. We expect that fine-tuning on CoLA results in changes in accuracy on a syntactic probing task.[2]

SST-2 The Stanford Sentiment Treebank (Socher et al., 2013). We use the binary version where the task is to categorize movie reviews to have either positive or negative valence. Making sentiment judgments requires knowing the meanings of isolated words and combining them on the sentence and discourse level (e.g. in case of irony). Hence, we expect to see a difference for semantic and/or discourse probing tasks when fine-tuning on SST-2.

RTE The Recognizing Textual Entailment dataset is a collection of sentence-pairs in either neutral or entailment relationship collected from a series of annual textual entailment challenges (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009). The task requires a deeper understanding of the relationship of two sentences, hence, fine-tuning on RTE might affect the accuracy on a discourse-level probing task.

SQuAD The Stanford Questions Answering Dataset (Rajpurkar et al., 2016) is a popular extractive reading comprehension dataset. The task involves a broader discourse understanding as a model trained on SQuAD is required to extract the answer to a question from an accompanying paragraph.

3.2 Probing Tasks

We select three sentence-level probing tasks from the SentEval probing suit (Conneau et al., 2018), testing for syntactic, semantic and broader discourse information on the sentence-level.

bigram-shift is a syntactic binary classification task that tests a model's sensitivity to word order. The dataset consists of intact and corrupted sentences, where for corrupted sentences, two random adjacent words have been inverted.

[1] Merchant et al. (2020) follow a similar reasoning. They find that fine-tuning on dependency parsing task leads to an improvement on the constituents probing task and attribute this to the improved linguistic knowledge. Similarly, Pruksachatkun et al. (2020) view probing task performance as "an indicator for the acquisition of a particular language skill."

[2] CoLA contains sentences with syntactic, morphological and semantic violations. However, only about 15% of the sentences are labeled with morphological and semantic violations. Hence, we suppose that fine-tuning on CoLA should increase a model's sensitivity to syntactic violations to a greater extent.

semantic-odd-man-out tests a model's sensitivity to semantic incongruity on a collection of sentences where random verbs or nouns are replaced by another verb or noun.

coordination-inversion is a collection of sentences made out of two coordinate clauses. In half of the sentences, the order of the clauses is inverted. Coordinate-inversion tests for a model's broader discourse understanding.

3.3 Pre-trained Models

It is unclear to which extent findings on the encoding of certain linguistic phenomena generalize from one pre-trained model to another. Hence, we examine three different pre-trained encoder models in our experiments.

BERT (Devlin et al., 2019) is a transformer-based model (Vaswani et al., 2017) jointly trained on masked language modeling and next-sentence-prediction – a sentence-level binary classification task. BERT was trained on the Toronto Books corpus and the English portion of Wikipedia. We focus on the **BERT-base-cased** model which consists of 12 hidden layers and will refer to it as BERT in the following.

RoBERTa (Liu et al., 2019b) is a follow-up version of BERT which differs from BERT in a few crucial aspects, including using larger amounts of training data and longer training time. The aspect that is most relevant in the context of this work is that RoBERTa was pre-trained without a sentence-level objective, minimizing only the masked language modeling objective. As with BERT we will consider the base model, **RoBERTa-base**, for this study and refer to it as RoBERTa.

ALBERT (Lan et al., 2020) is another recently proposed transformer-based pre-trained masked language model. In contrast to both BERT and RoBERTa, it makes heavy use of parameter sharing. That is, ALBERT ties the weight matrices across all hidden layers effectively applying the same non-linear transformation on every hidden layer. Additionally, similar to BERT, ALBERT uses a sentence-level pre-training task. We will use the base model **ALBERT-base-v1** and refer to it as ALBERT throughout this work.

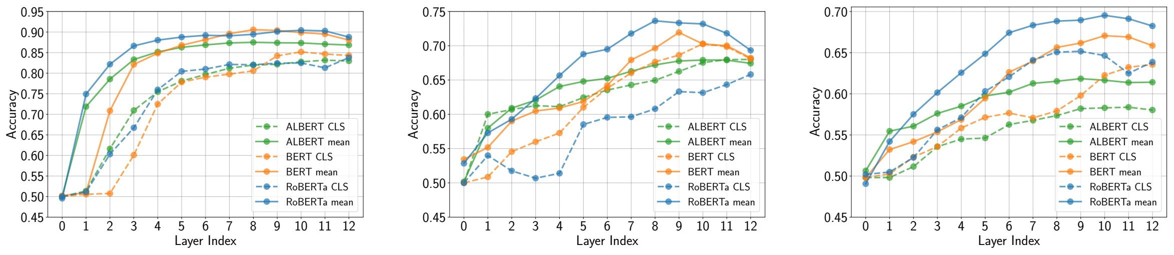
3.4 Fine-tuning and Probing Setup

Fine-tuning For fine-tuning, we follow the default setup proposed by Devlin et al. (2019). A single randomly initialized task-specific classification layer is added on top of the pre-trained encoder. As input, the classification layer receives $\mathbf{z} = \tanh(\mathbf{W}\mathbf{h} + \mathbf{b})$, where $\mathbf{h}$ is the hidden representation of the first token on the last hidden layer and $\mathbf{W}$ and $\mathbf{b}$ are the randomly initialized parameters of the classifier.[3] During fine-tuning all model parameters are updated jointly. We train for 3 epochs on CoLA and for 1 epoch on SST-2, using a learning rate of $2\mathrm{e}{-5}$. The learning rate is linearly increased for the first 10% of steps (warmup) and kept constant afterwards. An overview of all hyper-parameters for each model and task can be found in the Appendix. Fine-tuning performance on the development set of each of the tasks can be found in Table 1.

Probing For probing, our setup largely follows that of previous works (Tenney et al., 2019b; Liu et al., 2019a; Hewitt and Liang, 2019) where a *probing classifier* is trained on top of the contextualized embeddings extracted from a pre-trained or – as in our case – fine-tuned encoder model. Notably, we train *linear* (logistic regression) probing classifiers and use two different *pooling methods* to obtain sentence embeddings from the encoder hidden states: **CLS-pooling**, which simply returns the hidden state corresponding to the first token of the sentence and **mean-pooling** which computes a sentence embedding as the mean over all hidden states. We do this to assess the extent to which the CLS token captures sentence-level context. We use linear probing classifiers because intuitively we expect that if a linguistic feature is useful for a fine-tuning task, it should be linearly separable in the embeddings. For all probing tasks, we measure layer-wise accuracy to investigate how the linear separability of a particular linguistic phenomenon changes across the model. In total, we train 390 probing classifiers on top of 12 pre-trained and fine-tuned encoder models.

Implementation Our experiments are implemented in PyTorch (Paszke et al., 2019) and we use the pre-trained models provided by the HuggingFace transformers library (Wolf et al., 2019). Code to reproduce our results and figures is available online: `https://github.com/uds-lsv/probing-and-finetuning`

[3] For BERT and ALBERT $\mathbf{h}$ corresponds to the hidden state of the [CLS] token. For RoBERTa the first token of every sentence is the <s> token. We will refer to both of them as CLS token.



(a) bigram-shift

(b) coordination-inversion

(c) odd-man-out

Figure 1: Layer-wise probing accuracy on bigram-shift, coordination inversion, and odd-man-out for BERT, RoBERTa, and ALBERT. For all models mean-pooling (solid lines) consistently improves probing accuracy compared to CLS-pooling (dashed-lines) highlighting the importance of sentence-level information for each of the tasks.

Probing Task	BERT-base-cased							
	CLS-pooling				mean-pooling			
	CoLA		SST-2		CoLA		SST-2	
	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12
bigram-shift	0.07	4.73	−1.02	−4.63	0.23	1.45	−0.37	−3.24
coordinate-inversion	−0.10	1.90	−0.25	−1.15	0.14	0.29	−0.48	−0.85
odd-man-out	−0.20	0.26	−0.02	−1.28	−0.34	−0.29	−0.30	−1.09

Probing Task	RoBERTa-base							
	CLS-pooling				mean-pooling			
	CoLA		SST-2		CoLA		SST-2	
	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12
bigram-shift	0.58	5.35	−2.41	−7.22	0.69	1.74	−0.23	−4.87
coordinate-inversion	−0.72	1.84	−1.28	−0.63	−0.22	0.02	−0.18	−3.83
odd-man-out	−0.66	1.05	−1.09	−2.40	−0.08	−0.55	−0.46	−3.61

Probing Task	ALBERT-base-v1							
	CLS-pooling				mean-pooling			
	CoLA		SST-2		CoLA		SST-2	
	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12
bigram-shift	1.55	3.39	−1.94	−5.15	0.26	0.66	−0.70	−2.73
coordinate-inversion	−0.69	−1.53	−1.07	*-2.87*	−0.07	−1.19	−0.35	−1.53
odd-man-out	−0.42	−1.39	−0.90	−2.75	−0.27	−1.40	−0.60	−2.82

Table 2: Change in probing accuracy Δ (in $\%$) of **CoLA** and **SST-2** fine-tuned models compared to the pre-trained models when using CLS and mean-pooling. We average the difference in probing accuracy over two different layers groups: layers 0 to 6 and layers 7 to 12.

4 Experiments

4.1 Probing Accuracy

Figure 1 shows the layer-wise probing accuracy of BERT, RoBERTa, and ALBERT on each of the probing tasks. These results establish baselines for our comparison with fine-tuned models below. Consistent with previous work (Krasnowska-Kieraś and Wróblewska, 2019), we observe that mean-pooling generally outperforms CLS-pooling across all probing tasks, highlighting the importance of sentence-level context for each of the probing tasks. We also find that for *bigram-shift* probing accuracy is substantially larger than that for coordination-inversion and odd-man-out. Again, this is consistent with findings in previous works (Tenney et al., 2019b; Liu et al., 2019a; Tenney et al., 2019a) reporting better performance on syntactic than semantic probing tasks.

When comparing the three encoder models, we observe some noticeable differences. On *odd-man-out*, ALBERT performs significantly worse than both BERT and RoBERTa, with RoBERTa performing best across all layers. We attribute the poor performance of ALBERT to the fact that it makes heavy use of weight-sharing, effectively applying the same non-linear transformation on all layers. We also observe that on *coordination-inversion*, RoBERTa with CLS pooling performs much worse than both BERT and ALBERT with CLS pooling. We attribute this to the fact that RoBERTa lacks a sentence-level pre-training objective and the CLS token hence fails to capture relevant sentence-level information for this particular probing task. The small differences in probing accuracy for BERT and ALBERT when comparing CLS to mean-pooling and the fact that RoBERTa with mean-pooling outperforms all other models on *coordination-inversion* is providing evidence for this interpretation.

4.2 How does Fine-tuning affect Probing Accuracy?

Having established baselines for the probing accuracy of the pre-trained models, we now turn to the question of how it is affected by fine-tuning. Table 2 shows the effect of fine-tuning on CoLA and SST-2 on the layer-wise accuracy for all three encoder models across the three probing tasks. Results for RTE and SQuAD can be found in Table 5 in the Appendix. **For all models and tasks we find that fine-tuning has mostly an effect on higher layers, both positive and negative.** The impact varies depending on the fine-tuning/probing task combination and underlying encoder model.

Positive Changes in Accuracy: Fine-tuning on CoLA results in a substantial improvement on the *bigram-shift* probing task for all the encoder models; fine-tuning on RTE improves the *coordination-inversion* accuracy for RoBERTa. This finding is in line with our expectations: *bigram-shift* and CoLA require syntactic level information, whereas *coordination-inversion* and RTE require a deeper discourse-level understanding. However, when taking a more detailed look, this reasoning becomes questionable: The improvement is only visible when using CLS-pooling and becomes negligible when probing with mean-pooling. Moreover, the gains are not large enough to improve significantly over the mean-pooling baseline (as shown by the stars and the second y-axis in Figure 4). This suggests that adding new linguistic knowledge is not necessarily the *only* driving force behind the improved probing accuracy and we provide evidence for this reasoning in Section 5.1.

Negative Changes in Accuracy: Across all models and pooling methods, fine-tuning on SST-2 has a negative impact on probing accuracy on *bigram-shift* and *odd-man-out*, and the decrease in probing accuracy is particularly large for RoBERTa. Fine-tuning on SQuAD follows a similar trend: it has a negative effect on probing accuracy on *bigram-shift* and *odd-man-out* for both CLS- and mean-pooling (see Table 5), while the impact on *coordination-inversion* is negligible. We argue that this strong negative impact on probing accuracy is the consequence of more dramatic changes in the representations. We investigate this issue further in Section 5.2.

Changes in probing accuracy for other fine-tuning/probing combinations are not substantial, which suggests that representations did not change significantly with regard to the probed information.

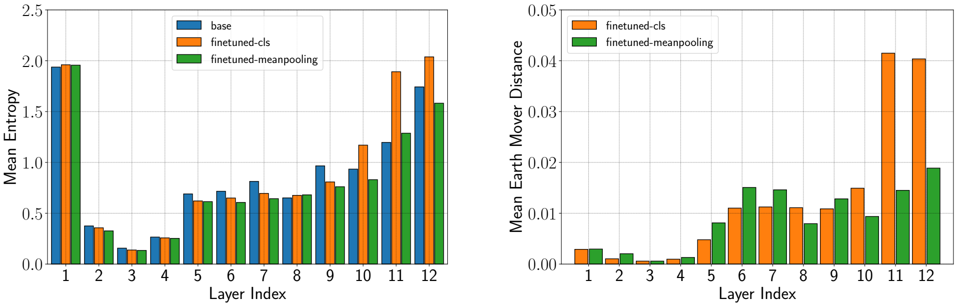
5 What Happens During Fine-tuning?

In the previous part, we saw the effects of different fine-tuning approaches on model performance. This opens the question for their causes. In this section, we study two hypotheses that go towards explaining these effects.

5.1 Analyzing Attention Distributions

If the improvement in probing accuracy with CLS-pooling can be attributed to a better sentence representation in the CLS token, this can be due to a corresponding change in a model's attention distribution. The model might change the attention of the CLS token to cover more tokens and with this build a better representation of the whole sentence.

To study this hypothesis, we fine-tune RoBERTa on CoLA using two different methods: the default CLS-pooling approach and mean-pooling (cf. Section 3.4). We compare the layer-wise attention



(a) Entropy

(b) Earth-Mover Distance

Figure 2: Entropy and Earth mover's distance of the attention for the CLS token for each layer with the RoBERTa model on the bigram-shift dataset. The mean over all input sequences and the mean over all attention heads of a layer are taken. The Earth Mover Distance is computed between the base model and each fine-tuned model.

distribution on bigram-shift after fine-tuning to that data. We expect to see more profound changes for CLS-pooling than for mean-pooling. To investigate how the attention distribution changes, we analyze its entropy, i.e.

$$H_j = \sum_i a_j(x_i) \cdot log\left(a_j(x_i)\right) \qquad (1)$$

where x_i is the i-th token of an input sequence and $a(x_i)$ the corresponding attention at position j given to it by a specific attention head. Entropy is maximal when the attention is uniform over the whole input sequence and minimal if the attention head focuses on just one input token.

Figure 2a shows the mean entropy for the CLS token (i.e. H_0) before and after fine-tuning. We observe a large increase in entropy in the last three layers when fine-tuning on the CLS token (orange bars). This is consistent with our interpretation that, during fine-tuning, the CLS token learns to take more sentence-level information into account, therefore being required to spread its attention over more tokens. For mean-pooling (green bars) this might not be required as taking the mean over all token-states could already provide sufficient sentence-level information during fine-tuning. Accordingly, there are only small changes in the entropy for mean-pooling, with the mean entropy actually decreasing in the last layer.

Entropy alone is, however, not sufficient to analyze changes in the attention distribution. Even when the amount of entropy is similar, the underlying attention distribution might have changed. Figure 2b, therefore, compares the attentions of an attention head for an input sequence before and after fine-tuning using *Earth mover's distance* (Rubner et al., 1998). We find that, similarly to the entropy results, changes in attention tend to increase with the layer number and again, the largest change of the attention distribution is visible for the first token for layer 11 and 12 when pooling on the CLS-token, while the change is much smaller for mean-pooling. This affirms our hypothesis that improvements in the fine-tuning with CLS-pooling can be attributed to a change in the attention distribution which is less necessary for the mean-pooling.

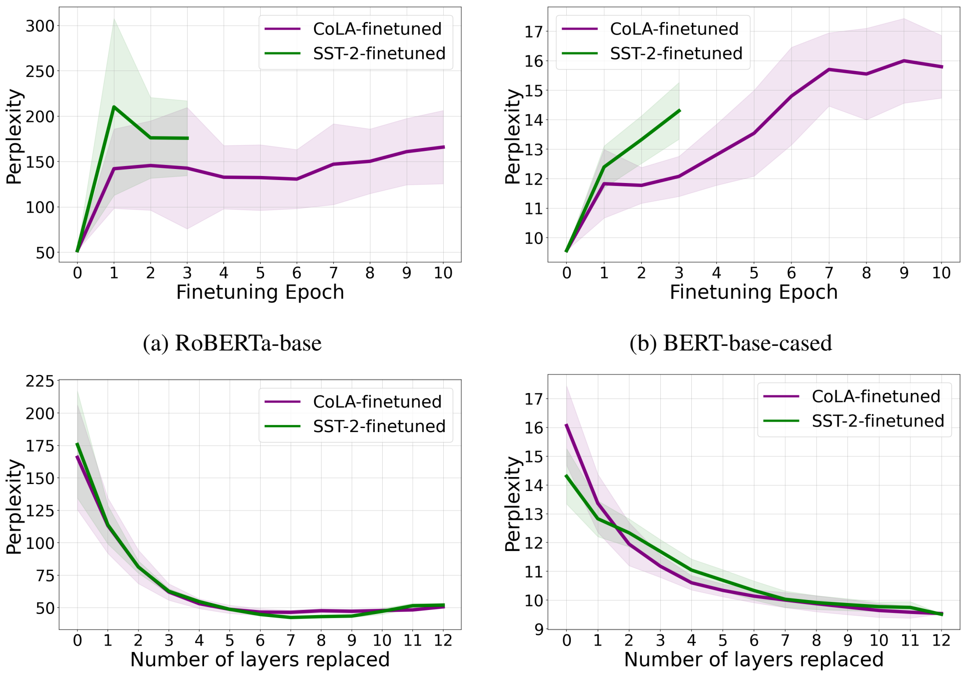
5.2 Analyzing MLM Perplexity

If fine-tuning has more profound effects on the representations of a pre-trained model potentially introducing or removing linguistic knowledge, we expect to see larger changes to the language modeling abilities of the model when compared to the case where fine-tuning just changes the attention distribution of the CLS token.

For this, we analyze how fine-tuning on CoLA and SST-2 affect the language modeling abilities of a pre-trained model. A change in perplexity should reveal if the representations of the model did change during fine-tuning and we expect this change to be larger for SST-2 fine-tuning where we observe a large negative increase in probing accuracy.

For the first experiment, we evaluate the pre-trained masked language model heads of BERT and RoBERTa on the Wikitext-2 test set (Merity et al., 2017) and compare it to the masked-language modeling perplexity, hereafter perplexity, of fine-tuned models.[4] In the second experiment, we test

[4] Note that perplexity results are not directly comparable between BERT and RoBERTa since both models have different vocabularies. However, what we are interested in is rather



(a) RoBERTa-base

(b) BERT-base-cased

(c) RoBERTa-base

(d) BERT-base-cased

Figure 3: Perplexity on Wikitext-2 of models consisting of a fine-tuned encoder and a pre-trained MLM-head. Plots (a) and (b) show how perplexity changes over the course of fine-tuning with epoch 0 showing the perplexity of the pre-trained model. (c) and (d) show how perplexity changes when a number of last layers of the fine-tuned encoder are replaced with corresponding layers from the pre-trained model. Note the different y-axes for RoBERTa and BERT.

which layers contribute most to the change in perplexity and replace layers of the fine-tuned encoder by pre-trained layers, starting from the last layer. For both experiments, we evaluate the perplexity of the resulting model using the pre-trained masked language modeling head. We fine-tune and evaluate each model 5 times, and report the mean perplexity as well as standard deviation. Our reasoning is that if fine-tuning leads to dramatic changes to the hidden representations of a model, the effects should be reflected in the perplexity.

Perplexity During Fine-tuning Figure 3a and 3b show how the perplexity of a pre-trained model changes during fine-tuning. Both BERT and RoBERTa show a similar trend where perplexity increases with fine-tuning. Interestingly, for RoBERTa the increase in perplexity after the first epoch is much larger compared to BERT. Additionally, our results show that for both models the increase in perplexity is larger when fine-tuning on SST-2. This confirms our hypothesis and also our findings from Section 4 suggesting that fine-tuning on SST-2 has indeed more dramatic effects on the representations of both models compared to fine-tuning on CoLA.

Perplexity When Replacing Fine-tuned Layers While fine-tuning leads to worse language modeling abilities for both CoLA and SST-2, it is not clear from the first experiment alone which layers are responsible for the increase in perplexity. Figure 3c and 3d show the perplexity results when replacing fine-tuned layers with pre-trained ones starting from the last hidden layer. Consistent with our probing results in Section 4, we find that the **changes that lead to an increase in perplexity happen in the last layers**, and this trend is the same for both BERT and RoBERTa. Interestingly, we observe no difference between CoLA and SST-2 fine-tuning in this experiment.

5.3 Discussion

In the following, we discuss the main implications of our experiments and analysis.

1. We conclude that fine-tuning indeed does affect the representations of a pre-trained model and in particular those of the last hidden layers, which is supported by our perplexity anal-

how perplexity changes with fine-tuning.

ysis. However, our perplexity analysis does not reveal whether these changes have a positive or negative effect on the encoding of linguistic knowledge.

2. Some fine-tuning/probing task combinations result in substantial improvements in probing accuracy when using CLS-pooling. Our attention analysis supports our interpretation that the improvement in probing accuracy can not simply be attributed to the encoding of linguistic knowledge, but can at least partially be explained by changes in the attention distribution for the CLS token. We note that this is also consistent with our findings that the improvement in probing accuracy vanishes when comparing to the mean-pooling baseline.

3. Some other task combinations have a negative effect on the probing task performance, suggesting that the linguistic knowledge our probing classifiers are testing for is indeed no longer (linearly) accessible. However, it remains unclear whether fine-tuning indeed removes the linguistic knowledge our probing classifiers are testing for from the representations or whether it is simply no longer linearly separable. We are planning to further investigate this in future work.

6 Conclusion

We investigated the interplay between fine-tuning and layer-wise sentence-level probing accuracy and found that fine-tuning can lead to substantial changes in probing accuracy. However, these changes vary greatly depending on the encoder model and fine-tuning and probing task combination. Our analysis of attention distributions after fine-tuning showed, that changes in probing accuracy can not be attributed to the encoding of linguistic knowledge alone but might as well be caused by changes in the attention distribution. At the same time, our perplexity analysis showed that fine-tuning has profound effects on the representations of a pre-trained model but our probing analysis can not sufficiently detail whether it leads to forgetting of the probed linguistic information. Hence we argue that the effects of fine-tuning on pre-trained representations should be carefully interpreted.

Acknowledgments

We thank Badr Abdullah for his comments and suggestions. We would also like to thank the reviewers for their useful comments and feedback, in particular R1. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project-id 232722074 – SFB 1102.

References

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

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

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.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09)*.

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. 2005. 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, page 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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

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

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, page 1–9, USA. 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.

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.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Katarzyna Krasnowska-Kieraś and Alina Wróblewska. 2019. Empirical linguistic study of sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5729–5739, Florence, Italy. Association for Computational Linguistics.

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

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to bert embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc.

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

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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*.

Y. Rubner, C. Tomasi, and L. J. Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. oLMpics–On what Language Model Pre-training Captures. *arXiv preprint arXiv:1912.13283*.

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, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

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

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Appendices

B Hyperparameters and Task Statistics

Table 3 shows hyperparamters used when fine-tuning BERT, RoBERTa, and ALBERT on CoLA, SST-2, RTE, and SQuAD. On SST-2 training for a single epoch was sufficient and we didn't observe a significant improvement when training for more epochs.

Table 4 shows number of training and development samples for each of the fine-tuning datasets considered in our experiments. Additionally, we report the metric used to evaluate performance for each of the tasks.

C Additional Results

Table 5 shows the effect of fine-tuning on RTE and SQuAD on the layer-wise accuracy for all three encoder models across the three probing tasks.

Figure 4 and Figure 5 show the change in probing accuracy Δ (in %) across all probing tasks

Hyperparameter	**Value**
Learning rate	2e−5
Warmup steps	10%
Learning rate schedule	warmup-constant
Batch size	32
Epochs	3 (1 for SST-2)
Weight decay	0.01
Dropout	0.1
Attention dropout	0.1
Classifier dropout	0.1
Adam ϵ	1e−8
Adam β_1	0.9
Adam β_2	0.99
Max. gradient norm	1.0

Table 3: Hyperparamters used when fine-tuning.

Statistics	**Task**			
	CoLA	SST-2	RTE	SQuAD
training	8.6k	67k	2.5	87k
validation	1,043	874	278	10k
metric	MCC	Acc.	Acc.	EM/F_1

Table 4: Fine-tuning task statistics.

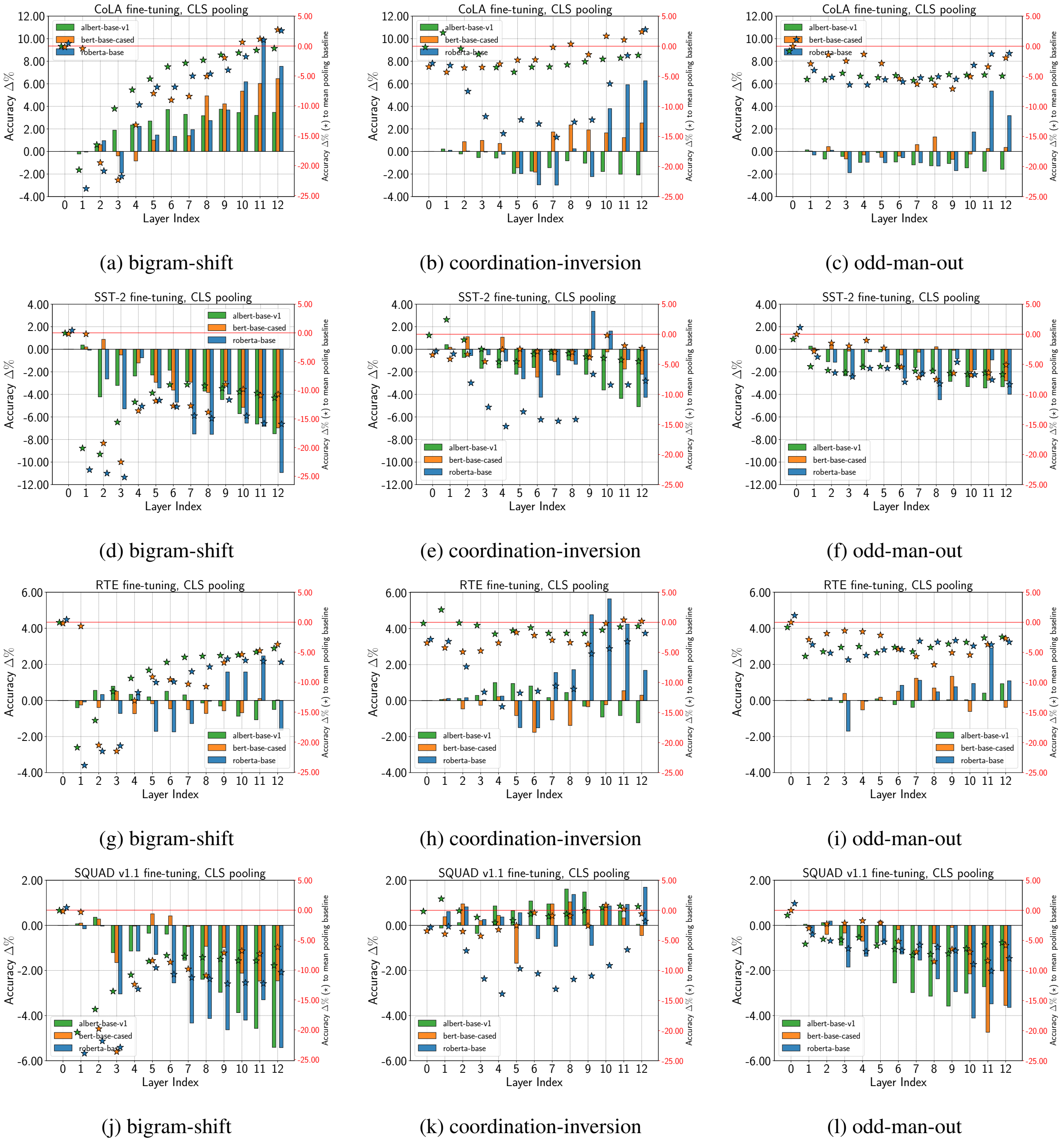
when fine-tuning on CoLA, SST-2, RTE, and SQuAD using CLS-pooling and mean-pooling, respectively. The second y-axis in Figure 4 shows the layer-wise difference after fine-tuning compared to the mean-pooling baseline. Note that only in very few cases this differences is larger than zero.x

Probing Task	**BERT-base-cased**							
	CLS-pooling				mean-pooling			
	RTE		SQuAD		RTE		SQuAD	
	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12
bigram-shift	−0.21	−0.39	−0.05	−1.50	−0.07	−0.31	−0.54	−1.66
coordinate-inversion	−0.43	−0.36	0.04	0.56	0.05	0.13	−0.03	0.10
odd-man-out	0.09	0.38	−0.21	−1.89	0.09	0.01	−0.28	−1.73

Probing Task	**RoBERTa-base**							
	CLS-pooling				mean-pooling			
	RTE		SQuAD		RTE		SQuAD	
	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12
bigram-shift	−0.51	0.44	−1.17	−4.33	−0.09	−1.32	−0.28	−3.09
coordinate-inversion	−0.35	3.27	0.29	0.50	0.30	−0.48	0.20	0.05
odd-man-out	−0.11	1.22	−0.76	−3.01	−0.04	−1.96	−0.21	−3.58

Probing Task	**ALBERT-base-v1**							
	CLS-pooling				mean-pooling			
	RTE		SQuAD		RTE		SQuAD	
	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12	0 – 6	7 – 12
bigram-shift	0.29	−0.43	−0.38	−3.46	−0.13	−0.82	−0.60	−3.11
coordinate-inversion	0.46	−0.44	0.32	0.92	0.13	−0.38	0.04	−0.27
odd-man-out	−0.03	0.17	−0.65	−2.91	−0.17	−0.85	−0.55	−3.18

Table 5: Change in probing accuracy Δ (in %) of **RTE** and **SQuAD** fine-tuned models compared to the pre-trained models when using CLS and mean-pooling. We average the difference in probing accuracy over two different layers groups: layers 0 to 6 and layers 7 to 12.



(a) bigram-shift

(b) coordination-inversion

(c) odd-man-out

(d) bigram-shift

(e) coordination-inversion

(f) odd-man-out

(g) bigram-shift

(h) coordination-inversion

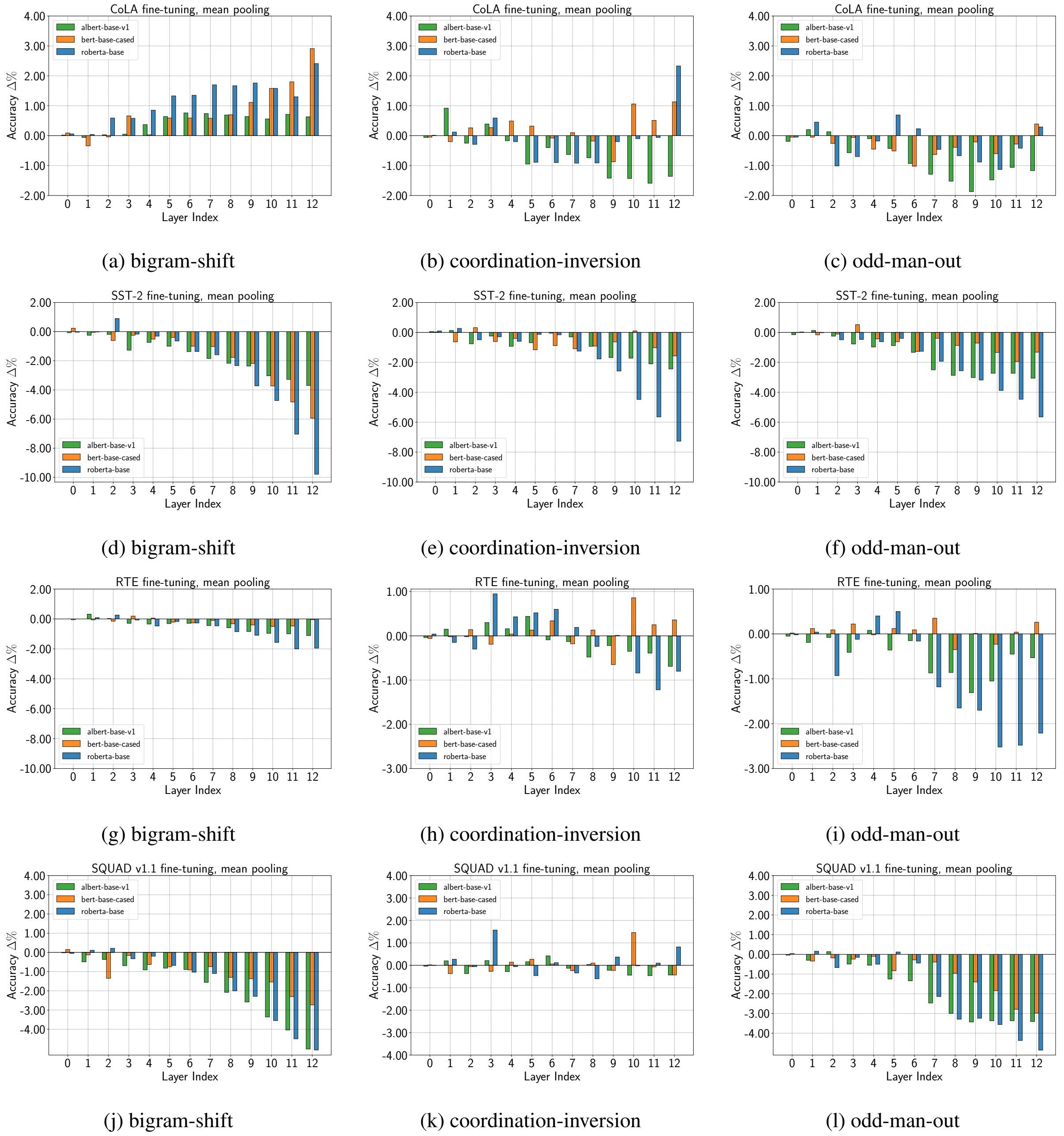
(i) odd-man-out

(j) bigram-shift

(k) coordination-inversion

(l) odd-man-out

Figure 4: Difference in probing accuracy Δ (in %) when using CLS-pooling after fine-tuning on **CoLA**, **SST-2**, **RTE**, and **SQuAD** for all three encoder models BERT, RoBERTa, and ALBERT across all probing taks considered in this work. The second y-axis shows layer-wise improvement over the mean-pooling baselines (stars) on the respective task.



(a) bigram-shift

(b) coordination-inversion

(c) odd-man-out

(d) bigram-shift

(e) coordination-inversion

(f) odd-man-out

(g) bigram-shift

(h) coordination-inversion

(i) odd-man-out

(j) bigram-shift

(k) coordination-inversion

(l) odd-man-out

Figure 5: Difference in probing accuracy Δ (in $\%$) when using mean-pooling after fine-tuning on **CoLA**, **SST-2**, **RTE**, and **SQuAD** for all three encoder models BERT, RoBERTa, and ALBERT across all probing tasks considered in this work.