How transfer learning impacts linguistic knowledge in deep NLP models?

Nadir Durrani **Hassan Sajjad** **Fahim Dalvi**
{ndurrani,hsajjad,faimaduddin}@hbku.edu.qa
Qatar Computing Research Institute, HBKU Research Complex, Doha 5825, Qatar

Abstract

Transfer learning from pre-trained neural language models towards downstream tasks has been a predominant theme in NLP recently. Several researchers have shown that deep NLP models learn non-trivial amount of linguistic knowledge, captured at different layers of the model. We investigate how fine-tuning towards downstream NLP tasks impacts the learned linguistic knowledge. We carry out a study across popular pre-trained models BERT, RoBERTa and XLNet using layer and neuron-level diagnostic classifiers. We found that for some GLUE tasks, the network relies on the core linguistic information and preserve it deeper in the network, while for others it forgets. Linguistic information is distributed in the pre-trained language models but becomes localized to the lower layers post-fine-tuning, reserving higher layers for the task specific knowledge. The pattern varies across architectures, with BERT retaining linguistic information relatively deeper in the network compared to RoBERTa and XLNet, where it is predominantly delegated to the lower layers.

1 Introduction

Contextualized word representations learned in transformer-based language models capture rich linguistic knowledge, making them ubiquitous for transfer learning towards downstream NLP problems such as Natural Language Understanding tasks e.g. GLUE (Wang et al., 2018). The general idea is to pretrain representations on large scale unlabeled data and adapt these towards a downstream task using supervision.

Descriptive methods in neural interpretability investigate what knowledge is learned within the representations through relevant extrinsic phenomenon varying from word morphology (Vylomova et al., 2016; Belinkov et al., 2017a; Dalvi et al., 2017) to high level concepts such as structure (Shi et al., 2016; Linzen et al., 2016) and semantics (Qian et al., 2016; Belinkov et al., 2017b) or more generic properties such as sentence length (Adi et al., 2016; Bau et al., 2019). These studies are carried towards analyzing representations from pre-trained models.[1] However, it is important to investigate how this learned knowledge evolves as the models are adapted towards a specific task from the more generic task of language modeling (Peters et al., 2018) that they are primarily trained on.

In this work, we analyze representations of 3 popular pre-trained models (BERT, RoBERTa and XLnet) with respect to morpho-syntactic and semantic knowledge, as they are fine-tuned towards GLUE tasks. More specifically we investigate i) if the fine-tuned models retain the same amount of linguistic information, ii) how this information is redistributed across different layers and individual neurons. To this end, we use *Diagnostic Classifiers* (Hupkes et al., 2018; Conneau et al., 2018), a popular framework for probing knowledge in neural models. The central idea is to extract feature representations from the network and train an auxiliary classifier to predict the property of interest. The quality of the trained classifier on the given task serves as a proxy to the quality of the extracted representations w.r.t to the understudied property (Belinkov et al., 2020).

We carry layer-wise (Liu et al., 2019a) and neuron-level probing analyses (Dalvi et al., 2019a) to study the fine-tuned representations. The former probes representations from individual layers w.r.t a linguistic property and the latter finds salient neurons in the network that capture the property. Fine-tuning involves adjusting feature weights, therefore it is important to look at the individual neurons to uncover important details, in addition to a more holistic layer-wise view.

[1] See recent surveys on representation analysis (Belinkov and Glass, 2019) and neuron analyses (Sajjad et al., 2021)

Our layer-wise analysis shows: i) that some GLUE tasks rely on core linguistic knowledge and the model preserves the information deeper in the network, while for others it is retained only in the lower layers ii) interesting cross-architectural differences with knowledge regressed to lower layers in RoBERTa and XLNet as opposed to BERT where it is still retained at the higher layers. Our neuron-wise analysis shows: i) salient linguistic neurons are relocated from the higher to lower layers, reinforcing our layer-wise results, ii) that linguistic information becomes less distributed and less redundant in the network post fine-tuning. Finally, we show how our analysis entails findings in layer pruning. Dropping higher layers of the models maintains comparable performance to tuning the full network, with linguistic information regressed to the lower layers. Conversely, pruning the lower layers (which hold the linguistic information) leads to substantial degradation in performance.

In comparison to the related work done in this direction, our findings resonate with Merchant et al. (2020) who found that fine-tuning primarily affects top layers and does not lead to "catastrophic forgetting of linguistic phenomena" in BERT. However, we found that other models like RoBERTa and XLNet, which they did not study, see a substantial drop in accuracy even at the lower layers and start forgetting linguistic knowledge much earlier in the network. In contrast to Mosbach et al. (2020), we study core-linguistic phenomena whereas their study is based on sentence level probing tasks. Differently from both, we carry out a fine-grained neuron analysis which sheds light on how neurons are distributed and relocated post fine-tuning. Our work complements their findings while extending the layer-wise analysis to core-linguistic tasks and additionally looking at the distribution and relocation of neurons after fine-tuning.

2 Methodology

Our methodology is based on the probing framework called as *Diagnostic Classifiers*. We train a classifier using the activations generated from the trained neural network as static features, towards the task of predicting a certain linguistic property. The underlying assumption is that if the classifier can predict the property, the representations implicitly encode this information. We train layer- and neuron-wise probes using logistic-regression classifiers. Formally, consider a pre-trained neural language model $\mathbf{M}$ with L layers: $\{l_1, l_2, \ldots, l_L\}$. Given a dataset $\mathbb{D} = \{w_1, w_2, ..., w_N\}$ with a corresponding set of linguistic annotations $\mathbb{T} = \{t_{w_1}, t_{w_2}, ..., t_{w_N}\}$, we map each word w_i in the data $\mathbb{D}$ to a sequence of latent representations: $\mathbb{D} \overset{\mathbb{M}}{\longmapsto} \mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. The model is trained by minimizing the following loss function:

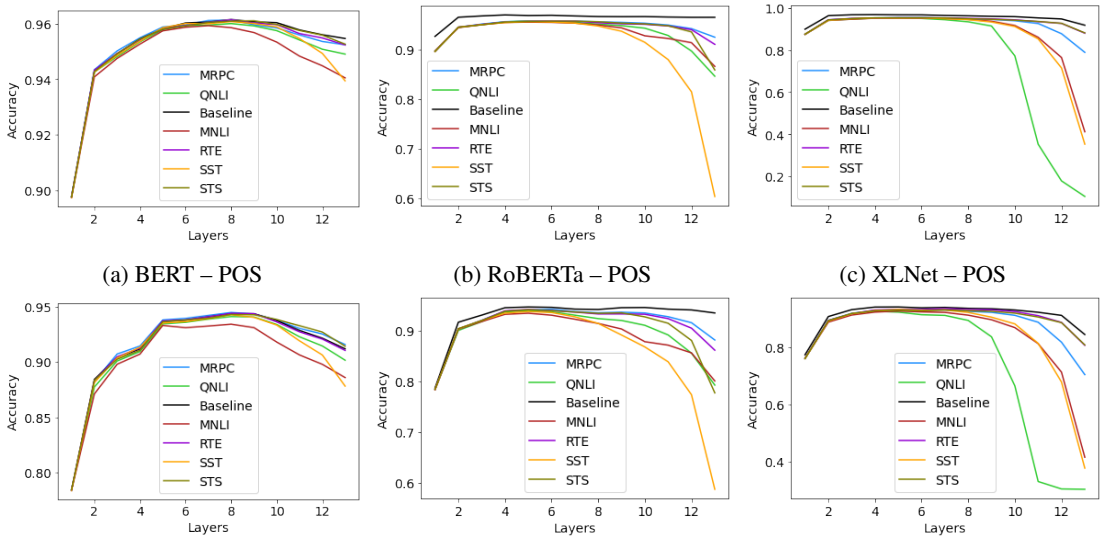
$$\mathcal{L}(\theta) = -\sum_i \log P_\theta(t_{w_i}|w_i) + \lambda_1\|\theta\|_1 + \lambda_2\|\theta\|_2^2$$

where $P_\theta(t_{w_i}|w_i) = \frac{\exp(\theta_l \cdot \mathbf{z}_i)}{\sum_{l'} \exp(\theta_{l'} \cdot \mathbf{z}_i)}$ is the probability that word i is assigned property t_{w_i}. We extract representations from the individual layers for our layer-wise analysis and the entire network for the neuron-analysis. We use the *Linguistic Correlation Analysis* as described in Dalvi et al. (2019a), to generate a neuron ranking with respect to the understudied linguistic property: Given the trained classifier $\theta \in \mathbb{R}^{D \times T}$, the algorithm extracts a ranking of the D neurons in the model $\mathbb{M}$ based on weight distribution. The elastic-net regularization (Zou and Hastie, 2005) – a combination of $\lambda_1\|\theta\|_1$ and $\lambda_2\|\theta\|_2^2$ is used to strike a balance between identifying focused ($L1$) versus distributed ($L2$) neurons. The weights for the regularization terms are tuned using a grid-search algorithm.

Following Durrani et al. (2020), we extract salient neurons for a linguistic property by iteratively choosing the top N neurons from the ranked list and retrain the classifier using these neurons, until the classifier obtains an accuracy close (within a specified threshold δ) to the *Oracle* – accuracy of the classifier trained using all the features in the network.

3 Experimental Setup

Pre-trained Neural Language Models: We experimented with 3 transformer models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b) and XLNet (Yang et al., 2019) using the base versions (13 layers and 768 dimensions). This choice of architectures leads to an interesting comparison between auto-encoder versus auto-regressive models. The models were then fine-tuned towards GLUE tasks of which we experimented with SST-2 for sentiment analysis with the Stanford sentiment treebank (Socher et al., 2013), MNLI for natural language inference (Williams et al., 2018), QNLI for Question NLI (Rajpurkar et al., 2016), RTE for recognizing textual entailment (Bentivogli et al., 2009), MRPC for Microsoft Research paraphrase



(a) BERT – POS (b) RoBERTa – POS (c) XLNet – POS

(d) BERT – Chunking (e) RoBERTa – Chunking (f) XLNet – Chunking

Figure 1: Layer-wise Probing Performance. Baseline refers to the performance of the pre-trained models without any finetuning.

corpus (Dolan and Brockett, 2005), and STS-B for the semantic textual similarity benchmark (Cer et al., 2017). The models were fine-tuned with the identical settings and we did 3 independent runs.

Linguistic Properties: We evaluated our method on 3 linguistic tasks: POS tagging using the Penn TreeBank (Marcus et al., 1993), syntactic chunking using CoNLL 2000 shared task dataset (Tjong Kim Sang and Buchholz, 2000), and semantic tagging using the Parallel Meaning Bank data (Abzianidze et al., 2017). We used standard splits for training, development and test data.

Classifier Settings: We used a linear probing classifier with elastic-net regularization, using a categorical cross-entropy loss, optimized by Adam (Kingma and Ba, 2014). Training is run with shuffled mini-batches of size 512 and stopped after 10 epochs. The regularization weights are trained using grid-search. For sub-word based models, we use the last activation value to be the representative of the word following Durrani et al. (2019). We computed *selectivity* (Hewitt and Liang, 2019) to ensure that our results reflect the property of representations and not the probe's capacity to memorize. Please see Appendix for details.

4 Analysis

4.1 Layer-wise Probing

First we train layer-wise probes to show how linguistic knowledge is redistributed across the network as we fine-tune it towards downstream tasks. Figure 1 shows results for POS and Chunking tasks.[2] We found varying observations across different GLUE tasks.

Comparing GLUE tasks: We found that linguistic phenomena are more important for certain downstream tasks, for example STS, RTE and MRPC where they are preserved in the higher layers post fine-tuning, as opposed to others, for example SST, QNLI and MNLI where they are forgotten in the higher layers. It would be interesting to study this further by connecting linguistic probes with any causation analysis on these tasks. Such an analysis would shed light on what concepts are used by the network while making predictions and why such information is forgotten for certain tasks. We leave this exploration for future.

Comparing Architectures: We found that pre-trained models behave differently in preserving information post fine-tuning. In the case of BERT, linguistic knowledge is fully preserved until layer 9, after which different task-specific models drop to varying degree, with SST and QNLI showing significant drop compared to others. An exception to this overall trend is MNLI where we start seeing a decline in performance earlier (between layers $5-7$). Contrastingly RoBERTa and XLNet show a depreciation in linguistic knowledge as early as layer 5. Also the drop is much more

[2]The observations are consistent for semantic tagging. Please see Appendix for results.

catastrophic in these two models with accuracy dropping by more than 35% in RoBERTa and 70% in XLNet. These results indicate that BERT retains its primarily learned linguistic knowledge and uses only a few of the final layers for fine-tuning, as opposed to XLNet and RoBERTa, where linguistic knowledge is retained only in the lower half of the network. Another cross-architectural observation that we made was that in RoBERTa and XLNet, the fine-tuned models do not ever reach the baseline performance (i.e. accuracy before fine-tuning – See Figure 1) at any layer, although the loss is $< 2\%$. We conjecture this discrepancy is due to the fact that the knowledge is more redundant and polysemous in the case of BERT, compared to XLNet, where it is more localized (also observed in Durrani et al. (2020)). Consequently, during fine-tuning XLNet and RoBERTa are more likely to lose linguistic information that is unimportant to the downstream task. We discuss this further in our neuron-analysis section.

4.2 Neuron-wise Probing

In our second set of experiments, we conducted analysis at a more fine-grained neuron level using *Linguistic Correlation Method* (Dalvi et al., 2019a). We extract the most salient neurons w.r.t a linguistic property (e.g. POS) and compare how the distribution of such neurons changes across the network as it is fine-tuned towards a downstream GLUE task. We use the weights of the trained classifier to rank neurons and select minimal set of salient neurons that give the same classifier accuracy as using the entire network in the baseline model. We found 5% neurons for POS and SEM tagging tasks and 10% for the Chunking tagging were sufficient to achieve the baseline performance.

Information becomes less distributed in the fine-tuned XLNet and RoBERTa models post fine-tuning: Table 1 shows accuracy of the classifier selecting the most (top) and least (bottom) 5% salient neurons on the task of POS tagging.[3] We observed that the bottom neurons in the fine-tuned models show a significant drop in performance, compared to the baseline model in the case of RoBERTa and XLNet. These results show that the information is more redundant in the baseline models as bottom neurons also preserved linguistic knowledge. On the contrary the information becomes more localized and less distributed in the fine-tuned models. The bottom neurons in the fine-tuned BERT changed the least, showing that linguistic information is still redundant and distributed in BERT.

Tasks	BERT		RoBERTa		XLNet	
	Top	Bot.	Top	Bot.	Top	Bot.
Base	96.0	94.9	96.7	95.3	96.5	91.2
MRPC	95.9	94.6	95.6	91.9	95.2	78.8
QNLI	96.0	94.6	95.8	84.3	94.7	10.3
MNLI	95.8	93.9	95.4	84.8	94.9	41.1
RTE	95.9	94.8	95.6	90.4	95.2	87.9
SST	95.9	94.2	95.6	60.4	95.0	35.2
STS	95.9	94.6	95.7	85.9	95.1	88.1

Table 1: POS accuracy – Top vs. Bottom neurons

How do salient neurons spread across the network layers? Previously we investigated how representations in each layer change w.r.t linguistic task. Now we study how the spread of the most salient neurons changes across the fine-tuned models. Figure 2 shows results for the selected GLUE tasks.[4] Notice how the most salient linguistic neurons shift from the higher layers towards the lower layers in RoBERTa and XLNet. This is especially pronounced in the case of *Roberta-SST* and *XLNet-QNLI* (See Figures 1e and 1f), where the number of salient chunking neurons significantly increased in the lower layers and droped in the higher layers, compared to the baseline. These findings reinforces our layer-wise results and additionally show how more responsibility is delegated to the neurons in the lower layers. Contrastingly, BERT did not exhibit this behavior. These results are inline with Durrani et al. (2020), who also found linguistic properties in XLNet to be localized to the lower layers[5] and fewer neurons and mutually exclusive as compared to BERT where neurons are highly polysemous[6] and therefore more redundant. Their finding helps us explain why XLNet forgets linguistic information that is unimportant to the downstream task more catastrophically.

5 Network Pruning

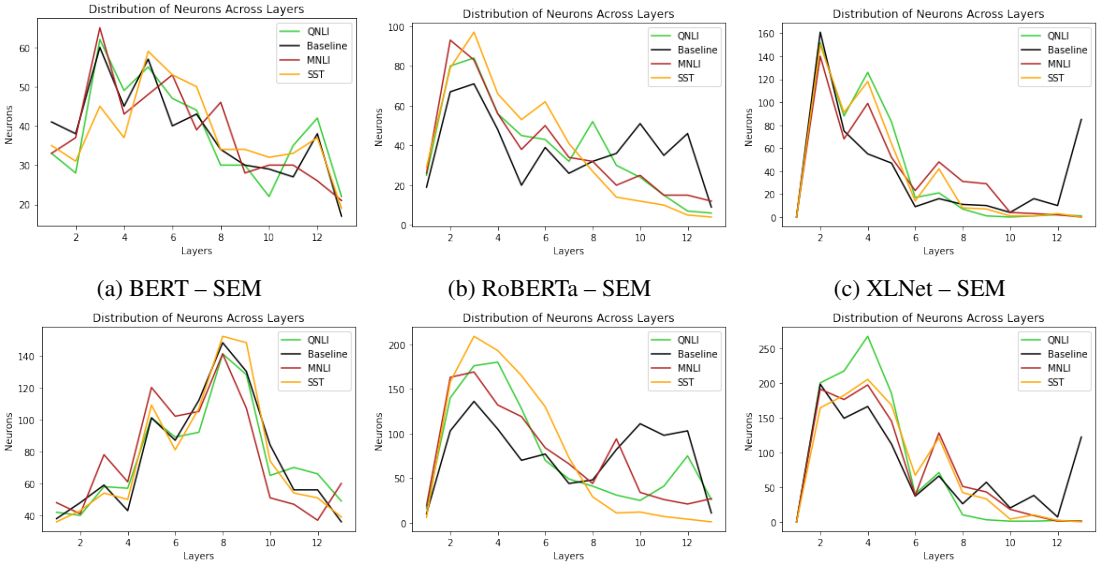
Our layer and neuron-wise analyses showed that core linguistic knowledge is redundant and distributed in the large pre-trained models. But as they are fine-tuned towards a down-stream task,

[3] See Appendix for SEM and Chunking tagging.

[4] See Appendix for all tasks and linguistic properties.

[5] Similarly (Wu et al., 2020) reported lower and middle layers of XLNet to have the most salient features.

[6] attend to multiple linguistic phenomenon



(a) BERT – SEM (b) RoBERTa – SEM (c) XLNet – SEM

(d) BERT – Chunking (e) RoBERTa – Chunking (f) XLNet – Chunking

Figure 2: Distribution of top neurons across layers

it is relocated and localized to lower layers, with higher layers focusing on the task-specific information. In this section, we show that our findings explain patterns in layer pruning. We question **How important is the linguistic knowledge for these downstream NLP tasks?** Following Sajjad et al. (2020) we prune top and bottom (excluding the embedding layer) 6 layers of the network in two separate experiments and compare architectures. Table 2 shows that removing bottom layers of the network in RoBERTa and XLNet leads to more damage compared to BERT. **How do these findings resonate with our analysis?** We showed that BERT retains linguistic information even at the higher layers of the model as opposed to RoBERTa where it is preserved predominantly at the lower layers. Removing the bottom 6 layers in RoBERTa leads to a bigger drop because the network is completely deprived of the linguistic knowledge. Linguistic knowledge is more distributed in BERT and preserved at the higher layers also which leads to a smaller drop as it can still access this information. We leave a detailed exploration on this for future.

Tasks	**SST**	**MNLI**	**QNLI**
		BERT	
Baseline	92.4	84.0	91.1
Prune Top 6	90.3	81.2	87.6
Prune Bottom 6	88.1	78.4	83.7
		RoBERTa	
Baseline	92.2	86.4	91.7
Prune Top 6	92.0	84.4	90.0
Prune Bottom 6	83.7	61.6	63.7
		XLNet	
Baseline	93.9	86.0	90.4
Prune Top 6	92.2	83.5	88.0
Prune Bottom 6	87.5	68.1	83.0

Table 2: Pruning Layers in the Models

6 Conclusion

We studied how linguistic knowledge evolves as the pre-trained language models are adapted towards downstream NLP tasks. We fine-tuned three popular models (BERT, RoBERTa and XLNet) towards GLUE benchmark and analyzed representations against core morpho-syntactic knowledge. We used probing classifiers to carry out layer and neuron-wise analyses. Our results showed that morpho-syntactic knowledge is preserved at the higher layers in some GLUE tasks (e.g. STS, MRPC and RTE), while forgotten and only retained at the lower layers in others (MNLI, QNLI and SST). Comparing architectures, we found that BERT retains linguistic knowledge deeper in the network. In the case of RoBERTa and XLNet, the information is only preserved in the middle layers. This discrepancy is due to the fact that neurons in BERT are more polysemous and distributed as opposed to XLNet and RoBERTa where they are more localized (towards lower layers) and mutually exclusive. We showed that this difference in architectures, entails different patterns as we prune top or bottom layers in the network. Our code is publicly as part of the NeuroX toolkit (Dalvi et al., 2019b).

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '17, pages 242–247, Valencia, Spain.

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*.

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

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

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46(1):1–52.

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

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*.

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*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and Improving Morphological Learning in the Neural Machine Translation Decoder. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*.

Fahim Dalvi, Avery Nortonsmith, D. Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. 2019b. NeuroX: A toolkit for analyzing individual neurons in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Honolulu, US.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. 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 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. 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.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

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.

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

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. On the Interplay Between Fine-tuning and Sentence-level Probing for Linguistic Knowledge in Pre-trained Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2502–2516, Online. Association for Computational Linguistics.

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.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Investigating Language Universal and Specific Properties in Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

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.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's bert: Smaller and faster transformer models.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021. Neuron-level Interpretation of Deep NLP Models: A Survey. *CoRR*, abs/2108.13138.

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*, EMNLP '16, pages 1526–1534, Austin, TX, USA.

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.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2016. Word Representation Models for Morphologically Rich Languages in Neural Machine Translation. *arXiv preprint arXiv:1606.04217*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.

A Appendices

A.1 Data and Representations

We used standard splits for training, development and test data for the 4 linguistic tasks (POS, SEM, Chunking) that we used to carry out our analysis on. The splits to preprocess the data are available through git repository[7] released with Liu et al. (2019a). See Table 3 for statistics. We obtained the understudied pre-trained models from the authors of the paper, through personal communication.

Task	Train	Dev	Test	Tags
POS	36557	1802	1963	44
SEM	36928	5301	10600	73
Chunking	8881	1843	2011	22

Table 3: Data statistics (number of sentences) on training, development and test sets using in the experiments and the number of tags to be predicted

A.2 Layer-wise Probing

Section 4.1 presented layer-wise probing results for POS and Chunking tagging. Figure 4 show results on Semantic tagging. We see a similar pattern across architectures as in Figure 1.

A.3 Neuron-wise Probing

Section 4.2 presented neuron-wise probing results for for Chunking tagging. Figure 2 show results on POS and SEM tagging. We see a similar pattern across architectures as in Figure 3. As the model is fine-tuned towards downstream, number of salient neurons towards a linguistic property, in the lower layers increase.

A.4 Top versus Bottom Neurons

In Section 4.2 we presented spread how information is more distributed and redundant in in the network as bottom neurons also preserved linguistic knowledge. On the contrary the linguistic information becomes more localized and less distributed post fine-tuning using accuracy of the bottom neurons. Tables 4 and 5 demonstrate the same pattern with respect to Chunking and Semantic tagging tasks, selecting 10% and 5% neurons respectively.

[7] https://github.com/nelson-liu/contextual-repr-analysis

Tasks	BERT		RoBERTa		XLNet	
	Top	Bot.	Top	Bot.	Top	Bot.
Base	94.7	92.3	94.8	92.5	94.2	92.3
MRPC	94.4	91.9	94.4	89.1	93.8	72.4
QNLI	94.3	92.3	94.0	82.2	93.0	33.3
MNLI	93.8	91.3	93.2	82.1	92.8	44.5
RTE	94.7	92.2	94.3	88.2	94.0	84.7
SST	94.3	91.9	94.1	60.7	93.8	39.7
STS	94.8	92.3	94.3	79.7	92.2	83.8

Table 4: Chunking accuracy – Top vs. Bottom neurons

Tasks	BERT		RoBERTa		XLNet	
	Top	Bot.	Top	Bot.	Top	Bot.
Base	92.2	90.9	92.8	90.7	96.5	91.2
MRPC	92.2	90.6	91.5	88.1	92.3	72.3
QNLI	92.0	90.8	91.5	78.6	91.4	17.8
MNLI	91.9	90.3	91.4	79.0	91.3	43.0
RTE	92.0	90.6	91.5	86.9	91.3	80.0
SST	92.1	90.5	91.5	60.7	91.3	34.8
STS	92.1	90.4	91.5	79.5	91.6	83.7

Table 5: SEM accuracy – Top vs. Bottom neurons

A.5 Pruning Layers

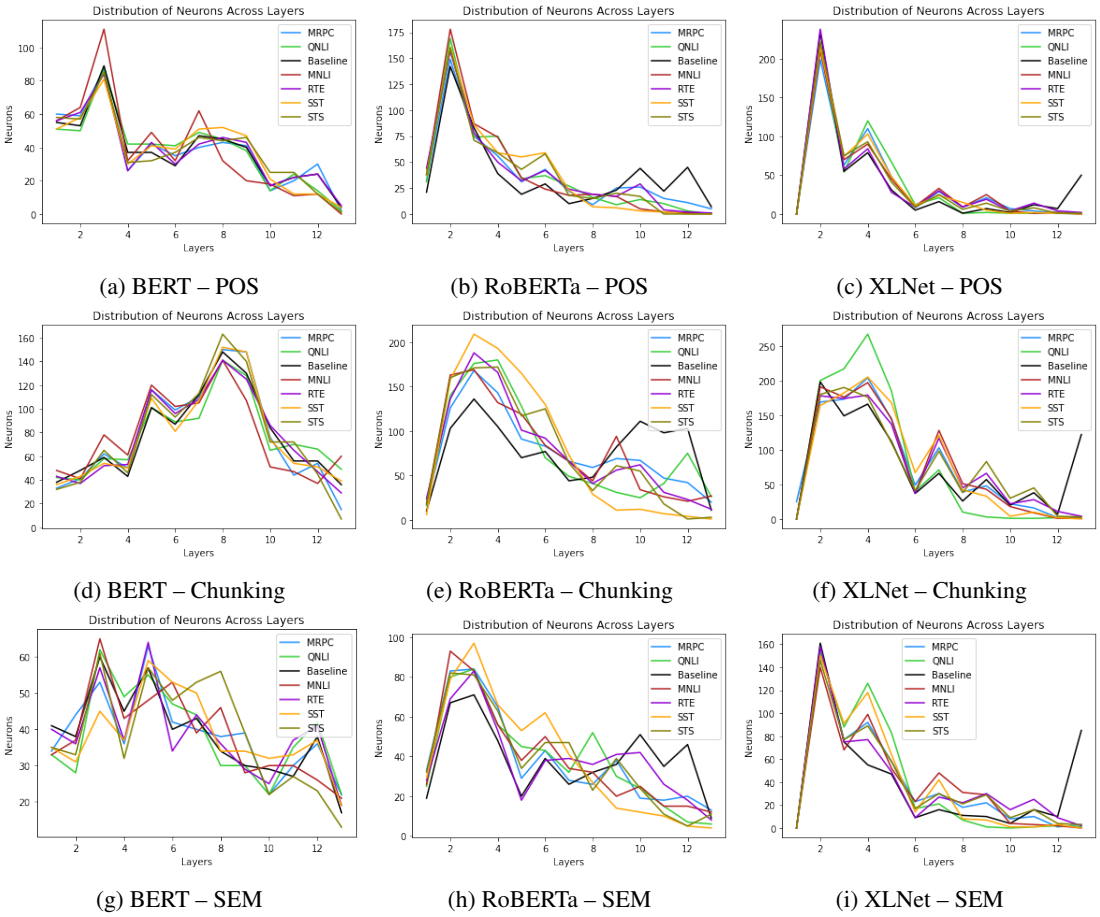
In Section 5 we showed how pruning bottom layers in RoBERTa was more harmful in comparison to BERT. We conjectured that this pattern entails from our analysis that in RoBERTa linguistic information is preserved in the initial middle layers as opposed to BERT where linguistic knowledge is distributed deeper in the network. We show that XLNet exhibit similar pattern to RoBERTa in Table 6.

A.6 Control Tasks

While there is a plethora of work demonstrating that contextualized representations encode a continuous analogue of discrete linguistic information, a question has also been raised recently if the representations actually encode linguistic structure or whether the probe memorizes the understudied task. We use *Selectivity* as a criterion to put a "linguistic task's accuracy in context with the probe's capacity to memorize from word types" (Hewitt and Liang,

Tasks	SST-2	MNLI	QNLI
		XLNet	
Baseline	93.9	86.0	90.4
Prune Top 6	92.2	83.5	88.0
Prune Bottom 6	87.5	68.1	83.0

Table 6: Pruning Layers in the Models



(a) BERT – POS (b) RoBERTa – POS (c) XLNet – POS

(d) BERT – Chunking (e) RoBERTa – Chunking (f) XLNet – Chunking

(g) BERT – SEM (h) RoBERTa – SEM (i) XLNet – SEM

Figure 3: Distribution of Top Neurons across Layers

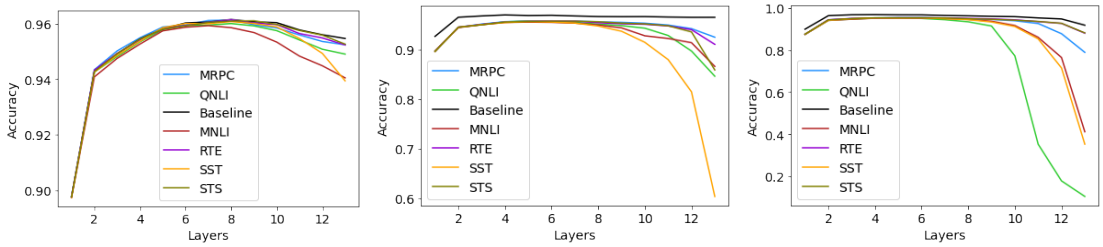
2019). It is defined as the difference between linguistic task accuracy and control task accuracy. An effective probe is recommended to achieve high linguistic task accuracy and low control task accuracy.

A.7 Infrastructure and Run Time

Our experiments were run on NVidia GeForce GTX TITAN X GPU card. Grid search for finding optimal lambdas is expensive when optimal number of neurons for the task are unknown. Running grid search would take $\mathcal{O}(MN^2)$ where $M = 100$ (if we try increasing number of neurons in each step by 1%) and $N = 0, 0.1, \ldots 1e^{-7}$. We fix the $M = 20\%$ to find the best regularization parameters first reducing the grid search time to $\mathcal{O}(N^2)$ and find the optimal number of neurons in a subsequent step with $\mathcal{O}(M)$. The overall running time of our algorithm therefore is $\mathcal{O}(M + N^2)$. This varies a lot in terms of wall-clock computation, based on number of examples in the training data, number of tags to be predicted in the downstream task. Including a full forward pass over the pre-trained model to extract the contextualized vector, and running the grid search algorithm to find the best hyperparameters and minimal set of neurons took on average 8 hours ranging from 3 hours for the Chunking experiment to 12 hours for POS and SEM due to large training data.

A.8 Hyperparameters

We use elastic-net based regularization to control the trade-off between selecting focused individual neurons versus group of neurons while maintaining the original accuracy of the classifier without any regularization. We do a grid search on L_1 and L_2 ranging from values $0 \ldots 1e^{-7}$. See Table 8 for the optimal values for each task across different architectures.



(a) BERT - SEM (b) RoBERTa - SEM (c) XLNet - SEM

Figure 4: Layer-wise Probing Performance

	BERT	XLNet	RoBERTa
Neu_a	9984	9984	9984
	POS		
Neu_t	500/5%	500/5%	500/5%
Acc_a	96.2	96.4	96.3
Acc_t	95.9	96.5	96.2
Sel_a	14.45	23.49	22.65
Sel_t	31.68	31.82	34.21
	SEM		
Neu_t	500/5%	500/5%	500/5%
Acc_a	92.51	92.29	92.95
Acc_t	92.32	92.62	92.97
Sel_a	5.77	14.03	13.76
Sel_t	27.17	26.55	24.53
	Chunking		
Neu_t	1000/10%	1000/10%	1000/10%
Acc_a	94.36	93.84	94.66
Acc_t	94.68	94.24	94.79
Sel_a	16.30	22.77	21.12
Sel_t	29.19	28.42	28.91

Table 7: Selecting minimal number of neurons for each downstream NLP task. Accuracy numbers reported on blind test-set (averaged over three runs) – Neu_a = Total number of neurons, Neu_t = Top selected neurons, Acc_a = Accuracy using all neurons, Acc_t = Accuracy using selected neurons after retraining the classifier using selected neurons, Sel = Difference between linguistic task and control task accuracy when classifier is trained on all neurons (Sel_a) and top neurons (Sel_t).

	BERT	XLNet	RoBERTa
	L1 , L2 = λ_1, λ_2		
POS	.001, .01	.001, .01	.001, .001
SEM	.001, .01	.001, .01	.001, .001
Chunk	$1e^{-4}, 1e^{-5}$	$1e^{-4}, 1e^{-4}$	.001, .001

Table 8: Best elastic-net lambdas parameters for each task