When BERT Plays the Lottery, All Tickets Are Winning

Sai Prasanna
Zoho
Estancia IT Park,
Chengalpattu, TN 603202, India
saiprasanna.r@zohocorp.com

Anna Rogers
Dept. of Computer Science,
Univ. of Massachusetts Lowell
Lowell, MA 01854
arogers@cs.uml.edu

Anna Rumshisky
Dept. of Computer Science,
Univ. of Massachusetts Lowell
Lowell, MA 01854
arum@cs.uml.edu

Abstract

Much of the recent success in NLP is due to the large Transformer-based models such as BERT (Devlin et al, 2019). However, these models have been shown to be reducible to a smaller number of self-attention heads and layers. We consider this phenomenon from the perspective of the lottery ticket hypothesis. For fine-tuned BERT, we show that (a) it is possible to find a subnetwork of elements that achieves performance comparable with that of the full model, and (b) similarly-sized subnetworks sampled from the rest of the model perform worse. However, the "bad" subnetworks can be fine-tuned separately to achieve only slightly worse performance than the "good" ones, indicating that most weights in the pre-trained BERT are potentially useful. We also show that the "good" subnetworks vary considerably across GLUE tasks, opening up the possibilities to learn what knowledge BERT actually uses at inference time.

1 Introduction

Much of the recent success in NLP is due to the transfer learning paradigm where large Transformer-based models first try to learn task-independent linguistic knowledge from large raw text corpora, and then get fine-tuned on small datasets for specific tasks. One of the most famous Transformers is BERT (Devlin et al., 2019), which became a must-have baseline and inspired dozens of analysis studies (Rogers et al., 2020b).

However, these models have been shown to be overparametrized. We now know that most Transformer heads and even layers can be pruned (Voita et al., 2019; Michel et al., 2019; Kovaleva et al., 2019), but it is not clear whether that is due to redundant weights, or to some parts of the model simply being "inactive" (Zhang et al., 2019).

We conduct a systematic case study of fine-tuning BERT (Devlin et al., 2019) on GLUE tasks (Wang et al., 2018) from the perspective of the lottery ticket hypothesis (Frankle and Carbin, 2019). We use importance scores for both self-attention heads and multi-layer-perceptrons (MLPs) in fine-tuned BERT to find the "good" subnetworks that achieve 90% of full model performance, and we test the lottery ticket hypothesis at the level of BERT architecture blocks. We find that "good" subnetworks perform considerably better than similarly-sized subnetworks sampled from the less important components of the model. However, both "bad" and "good" subnetworks can be fine-tuned separately to achieve comparable performance.

We also experiment with 9 GLUE tasks to see the degree to which the "good" subnetworks overlap. We find that 86% heads and 57% MLPs survive in less than 7 tasks, which raises concerns about the degree to which BERT relies on task-specific heuristics rather than general linguistic knowledge. It also offers a more precise instrument for learning what kinds of knowledge are used by BERT in different types of tasks and datasets.

2 Related work

BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019) has inspired multiple studies which aim to understand why it works so well and propose various modifications. A detailed overview of work to date is available in the survey by Rogers et al. (2020b).

One claim supported by many studies is that BERT is considerably overparametrized. In particular, it is possible to ablate elements of its architecture without loss in performance or even with slight gains (Michel et al., 2019; Voita et al., 2019; Kovaleva et al., 2019). This explains the success of BERT compression studies (Sanh et al., 2019; Jiao et al., 2019; McCarley, 2019; Lan et al., 2020).

While NLP focused on building larger Trans-

formers, the computer vision community was exploring the lottery ticket hypothesis (Frankle and Carbin, 2019; Lee et al., 2018; Zhou et al., 2019). It states that *"dense, randomly-initialized, feed-forward networks contain subnetworks (winning tickets) that – when trained in isolation – reach test accuracy comparable to the original network in a similar number of iterations"* (Frankle and Carbin, 2019). The "winning" initializations were shown to generalize across computer vision datasets (Morcos et al., 2019), and to exist both in LSTM and Transformer models for NLP tasks (Yu et al., 2020).

However, so far the lottery ticket work focused on the "winning" random initializations. In case of BERT and other widely used Transformers, there is a large pre-trained language model used in conjunction with a randomly initialized task-specific classifier. The motivation for this is that language modeling is a self-supervised task that can be performed on large amounts of text, and should yield transferable linguistic knowledge. The fine-tuning step would then only need to teach the model how to use the representation learned in pre-training to perform the specific task. However, we have ample evidence that BERT is very adapt at learning not just the new tasks, but also all kinds of biases present in the task-specific data (McCoy et al., 2019; Rogers et al., 2020a; Jin et al., 2020; Niven and Kao, 2019; Zellers et al., 2019).

An extra level of complexity is added by the fact that random initializations in the task-specific classifier interact with the pre-trained BERT weights, affecting the performance of fine-tuned BERT (Dodge et al., 2020). If the pre-trained weights indeed encode transferable linguistic knowledge, we would expect the "good" subnetworks to be the ones that better encode this knowledge, and they would be stable across different fine-tuning runs for the same task. The variation in performance between runs would then show that some initializations are better than others for leveraging the knowledge in the pre-trained weights for a given task. This is one of the questions we consider.

3 Methodology

The original lottery ticket study (Frankle and Carbin, 2019) focuses on feed-forward networks with iterative magnitude pruning. This section describes the alternative we use in the present study, namely, masking the "bad" subnetworks in BERT based on their importance scores.

3.1 Masking Heads and MLPs

BERT is fundamentally a stack of Transformer encoder layers (Vaswani et al., 2017). It consists of multiple identical layers, each containing several multi-head self-attention blocks followed by an MLP block with two residual connections.

The Multi-Head Self-Attention (**MHAtt**) consists of N_h independently parametrized self-attention heads. An attention head h in layer l is parametrized by $W_k^{(h,l)}, W_q^{(h,l)}, W_v^{(h,l)} \in \mathbb{R}^{d_h \times d}$, $W_o^{(h,l)} \in \mathbb{R}^{d \times d_h}$. d_h is typically set to d/N_h.

Given **n** d-dimensional input vectors $\mathrm{x} = x_1, x_2, ..x_n \in \mathbb{R}^d$ the multi-head attention is the sum of the output of each individual attention head applied to the input x.

$$\mathrm{MHAtt}^{(l)}(\mathrm{x}) = \sum_{h=1}^{N_h} \mathrm{Att}^{(l)}_{W_k^{(h,l)}, W_q^{(h,l)}, W_v^{(h,l)}, W_o^{(h,l)}}(\mathrm{x}) \quad (1)$$

The multi-layer perceptron MLP^l in layer l of BERT consists of two feed-forward layers. It is applied separately to **n** d-dimensional vectors $\mathrm{z} \in \mathbb{R}^d$ coming from the attention sub-layer. Dropout (Srivastava et al., 2014) is used for regularization. Then inputs of the MLP are added to its outputs through a residual connection.

$$\mathrm{MLP}^{(l)}_{\text{out}}(\mathrm{z}) = MLP^{(l)}(\mathrm{z}) + \mathrm{z} \quad (2)$$

For masking each self-attention head in a layer we change (1) to:

$$\mathrm{MHAtt}^{(l)}(\mathrm{x}) = \sum_{h=1}^{N_h} \xi^{(h,l)} \mathrm{Att}^{(l)}_{W_k^{(h,l)}, W_q^{(h,l)}, W_v^{(h,l)}, W_o^{(h,l)}}(\mathrm{x}) \quad (3)$$

where the $\xi^{(h,l)}$ are masking variables set to values $\{0, 1\}$. If we set $\xi^{(h,l)} = 0$ we effectively mask the attention head h in layer l.

For masking MLPs for a given layer we change (2) to:

$$\mathrm{MLP}^{(l)}_{\text{out}}(\mathrm{z}) = \nu^{(l)} MLP^{(l)}(\mathrm{z}) + \mathrm{z} \quad (4)$$

where the $\nu^{(l)}$ are masking variables set to values $\{0, 1\}$. If we set $\nu^{(l)} = 0$ we effectively mask the MLP in the layer l.

Task	Dataset	Train	Dev	Metric
CoLA	Corpus of Linguistic Acceptability Judgements (Warstadt et al., 2019)	10K	1K	Matthews
SST-2	The Stanford Sentiment Treebank (Socher et al., 2013)	67K	872	accuracy
MRPC	Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005)	4k	n/a	accuracy
STS-B	Semantic Textual Similarity Benchmark (Cer et al., 2017)	7K	1.5K	Pearson
QQP	Quora Question Pairs[1] (Wang et al., 2018)	400K	n/a	accuracy
MNLI	The Multi-Genre NLI Corpus (matched) (Williams et al., 2017)	393K	20K	accuracy
QNLI	Question NLI (Rajpurkar et al., 2016; Wang et al., 2018)	108K	11K	accuracy
RTE	Recognizing Textual Entailment (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009)	2.7K	n/a	accuracy
WNLI	Winograd NLI (Levesque et al., 2012)	706	n/a	accuracy

Table 1: GLUE tasks (Wang et al., 2018), dataset sizes and the metrics reported in this study

3.2 Importance scores

We mask the maximum number of attention heads and MLPs possible with the constraint that the model attains at least 90% of the performance of the full model. Combinatorial search to find this mask is impractical due to the compute required. Michel et al. (2019) proposed an importance score heuristic for self-attention heads in Transformers, which we adopt and extend to MLPs.

As a proxy score for component importance, we look at the expected sensitivity of the model to the mask variables $\xi^{(h,l)}$ in (3) and $\nu^{(l)}$ (4):

$$I_h^{(h,l)} = E_{x\sim X}\left|\frac{\partial \mathcal{L}(x)}{\partial \xi^{(h,l)}}\right| \quad (5)$$

$$I_{mlp}^{(l)} = E_{x\sim X}\left|\frac{\partial \mathcal{L}(x)}{\partial \nu^{(l)}}\right| \quad (6)$$

where x is a sample from the data distribution X and $\mathcal{L}(x)$ is the loss of the network outputs on that sample.

If $I_h^{(h,l)}$ and $I_{mlp}^{(l)}$ are high, they have a large effect on the model output. Absolute values are calculated to avoid highly positive contributions nullifying highly negative contributions.

In practice, calculating $I_h^{(h,l)}$ and $I_{mlp}^{(l)}$ would involve computing backward pass on the loss over samples of the evaluation data[2]. Following Michel et al., we normalize the importance scores for attention heads by layer (using the $\ell 2$ norm).

3.3 Iterative pruning

The importance scores described above are used iteratively to prune the lowest-scoring components of BERT. We continue pruning as long as the performance remains above 90% of the full fine-tuned model's performance. The components for pruning are selected under the following settings:

- *Heads only*: in each iteration, we mask as many of the unmasked heads with the lowest importance scores as we can (144 heads in the full BERT-base model).
- *MLPs only*: we iteratively mask one of the remaining MLPs that has the smallest importance score (Equation 5).
- *Heads and MLPs*: we compute head (Equation 5) and MLP (Equation 5) importance scores in a single backward pass, pruning 10% heads and one MLP with the smallest scores until the performance on the *dev* set is within 90%. Then we continue pruning heads alone, and then MLPs alone. This strategy results in a larger number of total components pruned within our performance threshold.

3.4 Fine-tuning

All experiments in this study are done on "BERT-base lowercase" pre-trained model, available in the Transformers library (Wolf et al., 2020). It is fine-tuned on 9 GLUE tasks, using the evaluation metrics shown in Table 1. All evaluation is done on the dev sets. For each experiment we test 5 random seeds.

Fine-tuning is performed with a modified GLUE script[3] of the Transformers library (v2.5.0). All parameters were set to their default values.

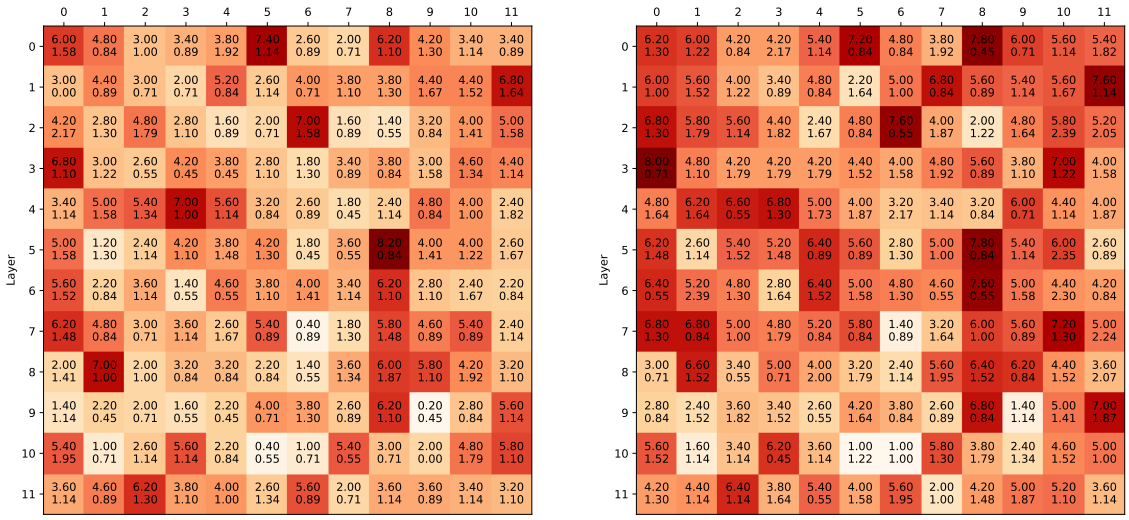
4 Experiments

4.1 The "good" subnetworks

This experiment follows up on the studies by Voita et al. (2019) that showed that only a few Trans-

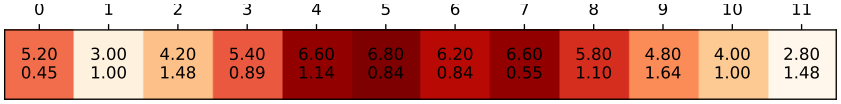
[2]The GLUE dev sets are used as oracles to obtain the best possible heads and MLPs for the particular model and task.

[3]https://github.com/huggingface/transformers/blob/v2.5.0/examples/run_glue.py

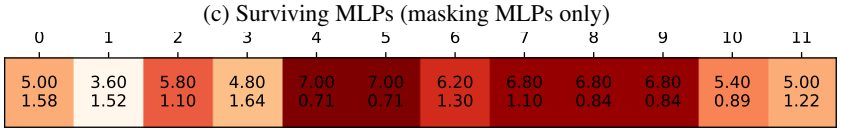


(a) Surviving heads (masking heads only)

(b) Surviving heads (masking heads and MLPs)



(c) Surviving MLPs (masking MLPs only)



(d) Surviving MLPs (masking heads and MLPs)

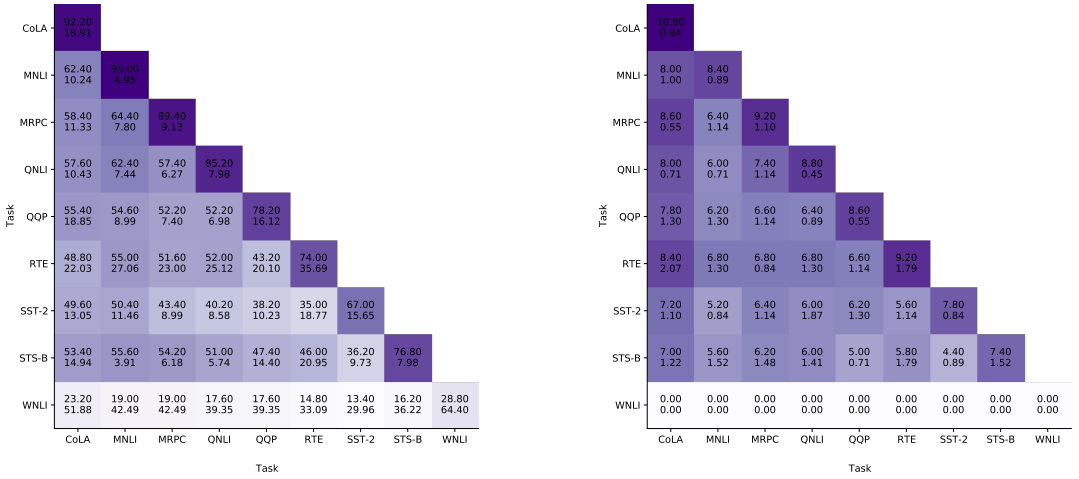
Figure 1: The "good" subnetworks: self-attention heads and MLPs that survive pruning. Each cell gives the average number of GLUE tasks in which a given head/MLP survived, and the standard deviation across 5 fine-tuning initializations.

former heads in machine translation task did the "heavy lifting", while the rest could be pruned. Michel et al. (2019) similarly showed that most of BERT self-attention heads in MNLI task could be pruned, and that the "good" heads were mostly shared between MNLI-matched and -mismatched. We extend this approach to 9 GLUE tasks, and we consider both BERT heads and MLPs.

We fine-tune BERT on each GLUE task with 5 random seeds, pruning elements of its architecture as described in section 3. We then compute how many times a given head survived the pruning process, for each task. Figure 1a and Figure 1b summarize the "good" subnetworks for individual tasks , showing the average number of GLUE tasks in which a given head survived, together with the standard deviation. We compare all pruning modes described in subsection 3.3: pruning only heads, only MLPs, and heads and MLPs together.

The subnetworks discovered in all pruning modes show a rather similar pattern of the useful heads and MLPs, but masking both heads and MLPs shows a larger number elements that survived in more than half the tasks (49% heads vs 22%, 75% MLPs vs 50%). This hints at considerable interaction between BERT's self-attention heads and MLPs. With fewer MLPs available the model is forced to rely more on the heads, raising their importance. This interaction was not explored in the previous studies focusing on only the heads or layers separately (Michel et al., 2019; Voita et al., 2019; Kovaleva et al., 2019), and deserves more attention in future work.

Similarly, BERT's MLPs also contain a "good" subnetwork (Figure 1c and Figure 1d). Here masking both heads and MLPs as opposed to only MLPs places more importance on the final layers of the model. Since self-attention heads change the most in the final layers of the model (Kovaleva et al., 2019), and those final self-attention heads do not affect the MLPs in the lower layers, it stands to reason that pruning MLPs and self-attention heads



(a) Heads shared between tasks

(b) MLPs shared between tasks

Figure 2: The "good" subnetwork: The diagonal represents the BERT architecture components that survive pruning for a given task and remaining elements represent the common surviving components across GLUE tasks. Each cell gives the average number of heads (out of 144) or layers (out of 12), together with standard deviation across 5 random initializations.

together makes the final layers more indispensable.

Note also that in both conditions the middle layers survive pruning for the majority of GLUE tasks. This is consistent with the findings by Liu et al. (2019) that the middle Transformer layers are the most transferable. K et al. (2020) also report that the depth of the model mattered more than the number of heads.

4.2 How task-independent are the "good" subnetworks?

Figure 1 shows that relatively few components of BERT survive pruning in most GLUE tasks. In the more lenient heads+MLPs pruning mode, only 7% heads and 17% MLPs survive in 7 out of 9 tasks, and could be interpreted as evidence of task-independent linguistic information.

Conversely, the parts of the "good" subnetworks that are only relevant for some specific tasks, but consistently survive across fine-tuning runs for that task, may correspond to task-specific information in the pretrained model – or possibly dataset-specific artifacts (Gururangan et al., 2018). Note that Figure 1 shows very few heads or MLPs that are universally "useless" (only 7 heads that survived in less than 2 tasks). 86% heads and 67% MLPs survive in 2-7 tasks with relatively high standard deviation. This means that *the "good" subnetworks for different tasks have relatively little in common*. The plots for all individual tasks are shown in Appendix A.

If most components of the "good" subnetwork are not universal across tasks, the degree to which the "good" subnetworks overlap across tasks may be a useful way to characterize the tasks themselves. This is illustrated in Figure 2, which shows pairwise comparisons between all GLUE tasks with respect to the number of shared surviving heads and MLPs in their "good" subnetworks (with standard deviation across 5 fine-tuning runs). The heads and MLPs were pruned together.

The results of this experiment say as much about BERT as about the target tasks. In particular, there is significant variation in standard deviations across tasks (shown in the diagonal cells in Figure 2a): only about 5 heads for MNLI, and 64 for WNLI. This comparative instability explains why WNLI results are so inconsistent[4]: the model cannot find a reliable signal in the pre-trained weights. Figure 2b shows that WNLI has zero overlaps with all tasks and itself because almost everything gets pruned (but the model performance actually goes up to the frequency baseline).

Interestingly, RTE also varies quite a bit in what heads and MLPs make it to the "good" subnetwork across runs, but that does not prevent BERT from reaching good results. That could mean that BERT

[4]The GLUE authors describe the dataset as "somewhat adversarial", with similar sentences in train and dev that have opposite labels.

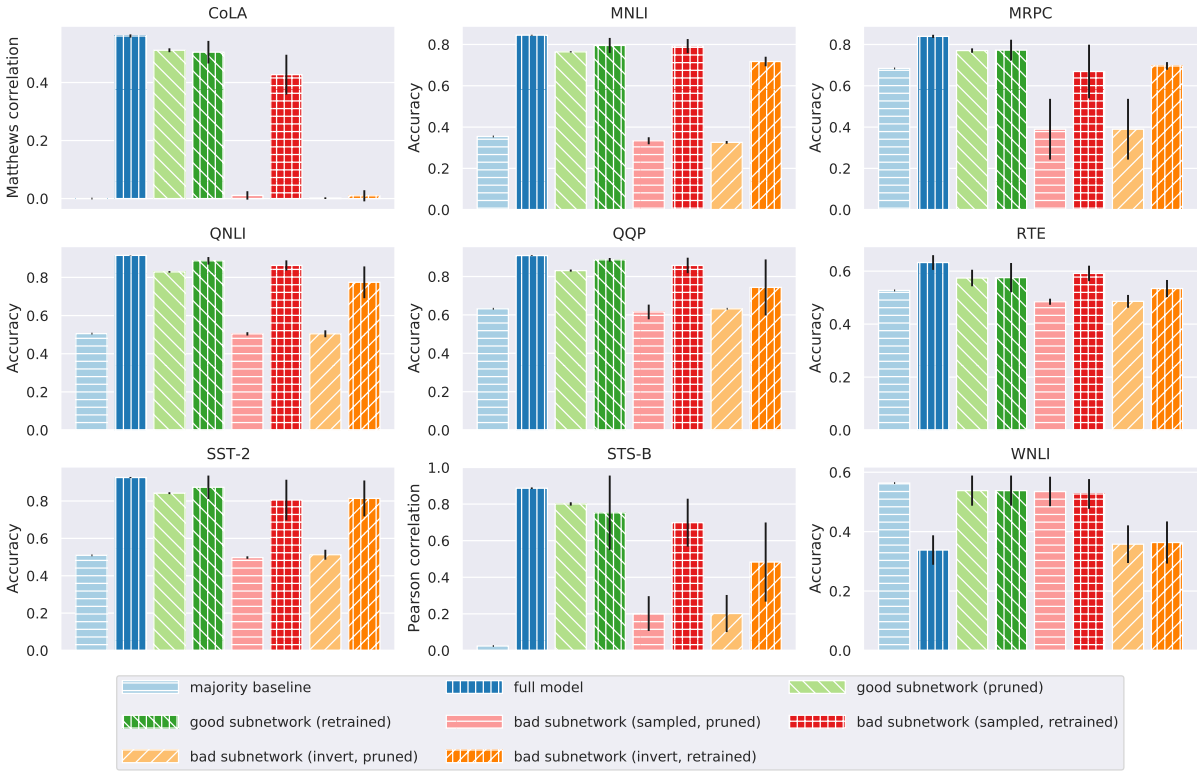


Figure 3: The "good" and "bad" subnetworks in BERT fine-tuning: performance on GLUE tasks (error bars indicate standard deviation across 5 fine-tuning runs).

provides several possible pathways for solving this task, all comparably good.

Based on the type of tasks, one could expect that SST would rely on different signal than NLI tasks, and that is indeed the case: after WNLI, SST has the least in common with the other tasks. However, the tasks focusing on similarity and paraphrase (MRPC, QQP, STS-B) and inference (MNLI, QNLI, RTE) are on par with each other. The two tasks that have the most in common with the others are MNLI (perhaps due to its multi-domain nature) and COLA (likely due to the variety of language phenomena it covered). Interestingly, these patterns are observed in both heads and MLPs, again pointing at the interaction between these components.

4.3 The "good" and "bad" subnetworks in BERT fine-tuning

Our final experiment puts the above evidence of "good" subnetworks in fine-tuned BERT from the perspective of lottery ticket hypothesis, which predicts that the "lucky" subnetworks can be re-trained from scratch to match the performance of the full network. To test this hypothesis, we experiment with the following subnetworks:

- *"good" subnetworks (pruned)*: the elements selected from the full model by importance scores, as described in subsection 3.3;
- *"bad" subnetworks (sampled)*: the elements sampled from those that did not survive the pruning, plus a random sample of elements with high importance scores so as to match the size of the "good" subnetworks;
- *"bad" subnetworks (pruned)*: simple inversion of the "good" subnetworks. They are 5-18% smaller in size than the sampled bad subnetworks, but they do not contain any elements with high importance scores.

For both pruned and sampled subnetworks we evaluate their performance on all tasks simply after pruning the full fine-tuned model, and with fine-tuning the same subnetwork with the same random seeds, with the rest of the model masked. The results of this experiment are shown in the Figure 3.

The main prediction of the lottery ticket hypothesis is validated: the "bad" subnetworks perform considerably worse that the "good" subnetworks if the rest of the model is pruned. This holds for both sampled and inverted "bad" subnetworks, al-

though the former include some "good" elements. The only task in which that does not hold is WNLI, the results of which are unreliable for reasons discussed above.

However, we see that both "good" and "bad" networks can be retrained, with comparable performance for many tasks. The inverted "bad" networks perform worse than the sampled ones, but that could also be due to them being smaller in size. Performance of all inverted "bad" networks on COLA is almost zero: since the "good" subnetwork comprises 92 of 144 heads and 10 out of 12 layers (Figure 2), very little remains when that mask is inverted.

5 Discussion

Does BERT have "bad" subnetworks? The key result of this study is that, as far as fine-tuning is concerned, BERT does not seem to have "bad" subnetworks that cannot be re-trained to relatively good performance level, suggesting that the weights that do not survive pruning are not just "inactive" (Zhang et al., 2019). However, it is important to remember that we consider elements of BERT architecture as atomic units, while the original lottery ticket work relied on magnitude pruning of individual weights. On that level BERT probably does have "bad" subnetworks: Yu et al. (2020) show that they can be found in MT Transformer models with global iterative pruning. We leave it to future research to find out to what extent the effective subnetworks overlap with the effective architectural blocks, and what that says about the architecture of BERT and other Transformers.

Our results suggest that most architecture blocks of BERT are potentially usable in fine-tuning, but this should *not* be interpreted as a proof that they all encode potentially relevant linguistic information. It is also possible that pre-training somehow simply made them more amenable to optimization, which is another question to future research.

What do the BERT components do for different tasks? Much of prior research explored the linguistic functions of individual BERT heads and layers (Htut et al., 2019; Clark et al., 2019; Lin et al., 2019; Vig and Belinkov, 2019; Hewitt and Manning, 2019; Goldberg, 2019; Tenney et al., 2019) with various probing tasks, but probing tasks do not show whether a piece of knowledge is actually used, even if it is present. Kovaleva et al. (2019) posed the question of whether a piece of linguistic knowledge that *should* be used by BERT *was* actually used at inference time, but were unable to confirm it for core frame-semantic relations.

The explorations of the "good" subnetworks of BERT elements, such as described in this paper, offer a fascinating direction for future research on the kinds of verbal reasoning that the model actually performs for a given task. We could find the "good" subnetworks and then look at its functions, rather than probe the whole model and hope that the knowledge found by the probes is actually used at inference time. We could also use the knowledge about which elements overlap in utility for different tasks to learn a lot more about the nature of transfer learning, as well as about specific tasks and datasets. For instance, consider the fact that the "good" subnetwork of MRPC shares many more heads with MNLI than with QQP or RTE, although they are closer by the type of the task (Figure 2a).

6 Conclusion

Prior work showed that it was possible to prune most self-attention heads in BERT. We extend this approach to the fully-connected layers, and we show fine-tuned BERT has "good" and "bad" subnetworks, where the "good" heads and MLPs alone reach performance comparable with the full network, and the "bad" ones do not perform well. However, this pattern does not quite conform to the lottery ticket hypothesis, as both "good" and "bad" networks can be fine-tuned separately to reach comparable performance.

We also show that 86% heads and 57% MLPs in "good" subnetworks are not universally useful across GLUE tasks, and overlaps between "good" subnetworks do not necessarily correspond to task types. This raises questions about the degree to which fine-tuned BERT relies on task-specific or general linguistic knowledge, and opens up the possibilities of studying the "good" subnetworks to see what types of knowledge BERT actually relies on at inference type.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.

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.

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.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

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.

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:2002.06305 [cs]*.

W.B. Dolan and C. Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing.

Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

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, pages 1–9, Prague, Czech Republic. Association for Computational Linguistics.

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

R. Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The Second Pascal Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

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.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in BERT track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. TinyBERT: Distilling BERT for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *AAAI 2020*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In *ICLR 2020*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4356–4365, Hong Kong, China. 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 *ICLR*.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2018. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 552–561.

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.

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.

JS McCarley. 2019. Pruning a BERT-based Question Answering Model. *arXiv preprint arXiv:1910.06360*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. 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.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? *Advances in Neural Information Processing Systems 32 (NIPS 2019)*.

Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. One ticket to win them all: Generalizing lottery ticket initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems 32*, pages 4932–4942. Curran Associates, Inc.

Timothy Niven and Hung-Yu Kao. 2019. Probing Neural Network Comprehension of Natural Language Arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. 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. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020a. Getting Closer to AI Complete Question Answering: A Set of Prerequisite Real Tasks. In *AAAI*, page 11.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020b. A Primer in BERTology: What we know about how BERT works. *arXiv:2002.12327 [cs]*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*.

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.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008, Long Beach, CA, USA.

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, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. 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.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. 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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtow-

icz, and Jamie Brew. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. 2020. Playing the lottery with rewards and multiple languages: Lottery tickets in RL and NLP. In *ICLR*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *ACL 2019*.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. 2019. Are All Layers Created Equal? In *ICML 2019 Workshop Deep Phenomena*.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. 2019. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask. In *Advances in Neural Information Processing Systems 32*, pages 3597–3607. Curran Associates, Inc.

A "Good" subnetworks in BERT fine-tuned on GLUE tasks

Each figure in this section shows the "good" subnetwork of heads and layers that survived the pruning process described in section 3. Each task was run with 5 different random seeds. The top number in each cell indicates how likely a given head or MLP was to survive pruning, with 1.0 indicating that it survived on every run. The bottom number indicates the standard deviation across runs.

The figures in this appendix show that each task has a varying number of heads and layers that survive pruning on all fine-tuning runs, while some heads and layers were only "picked up" by some random seeds. Note also that in addition to the architecture elements that survive across many runs, there are also those that are useful for over half of the tasks, as shown in Figure 1. Presumably they encode the most general linguistic information.

Note how visualizing the "good" subnetwork illustrates the core problem with WNLI, the most difficult task of GLUE. Figure 12 shows that each run is completely different, indicating that BERT fails to find any consistent pattern between the task and the information in the available pre-trained weights.

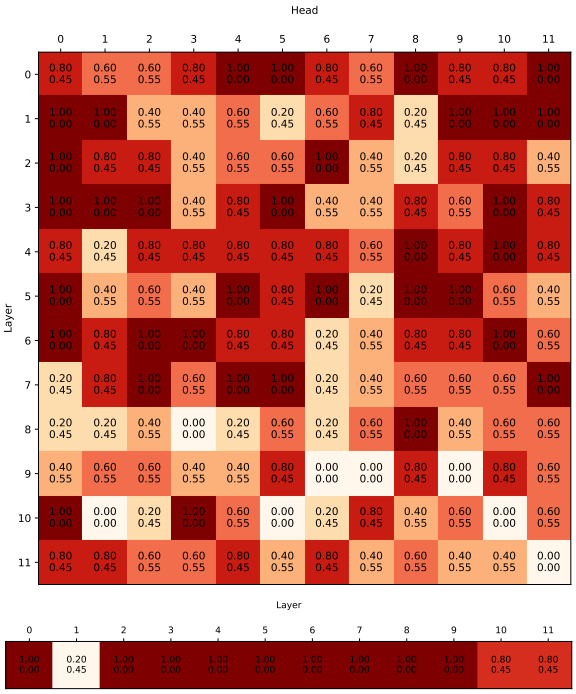


Figure 4: COLA

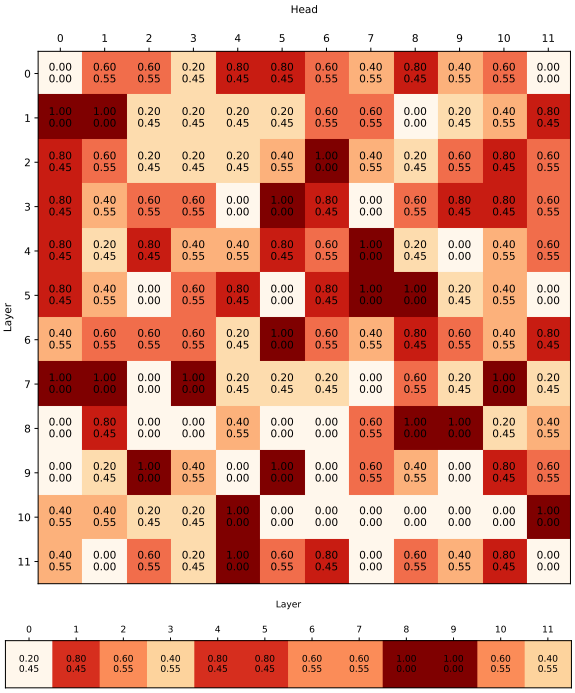


Figure 5: SST-2

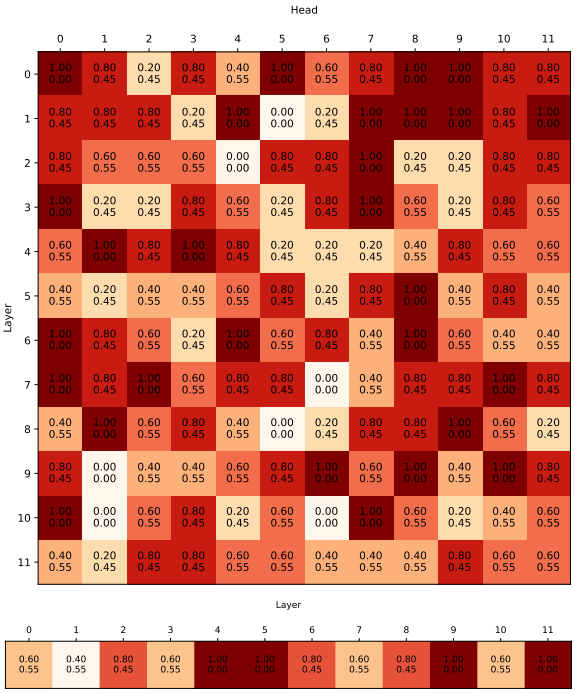


Figure 6: MRPC

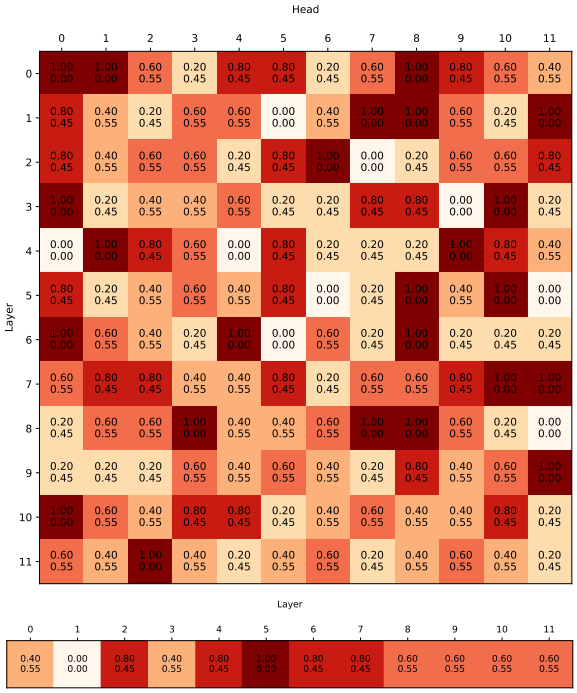


Figure 7: STS-B

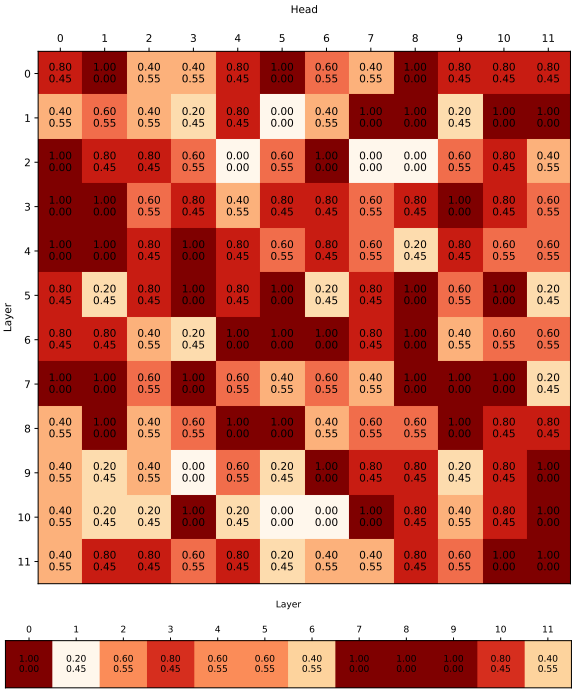


Figure 9: MNLI

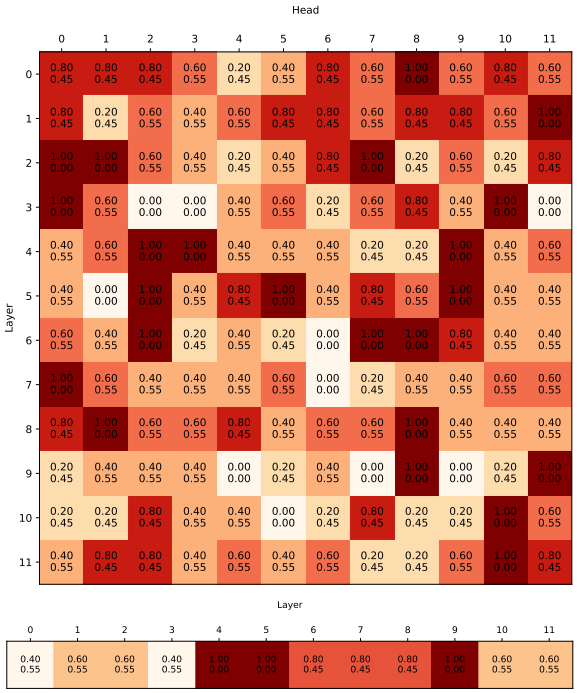


Figure 8: QQP

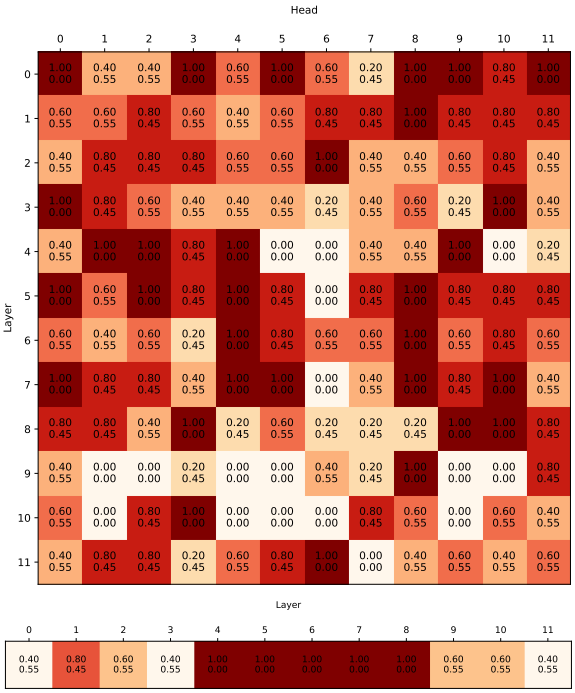


Figure 10: QNLI

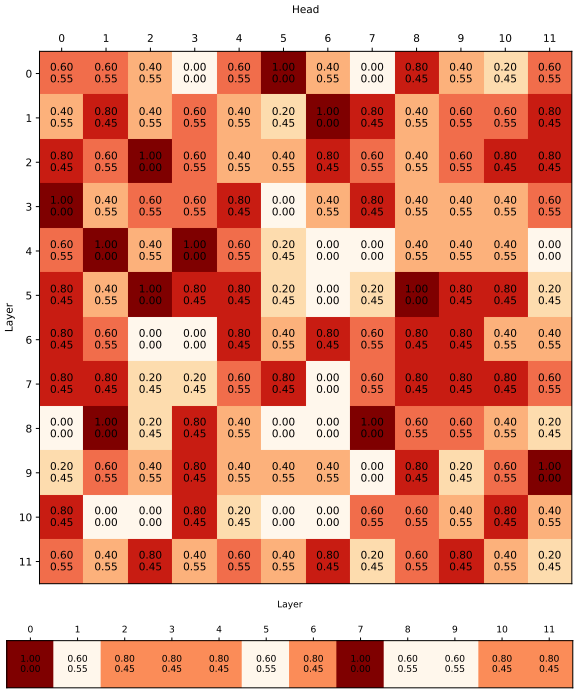


Figure 11: RTE

Figure 12: WNLI