How Much Does Attention Actually Attend? Questioning the Importance of Attention in Pretrained Transformers

Michael Hassid[♡] **Hao Peng**[♢*] **Daniel Rotem**[♡] **Jungo Kasai**[♠] **Ivan Montero**[★*]
Noah A. Smith[♠♢] **Roy Schwartz**[♡]
[♡]School of Computer Science & Engineering, Hebrew University of Jerusalem
[♢]Allen Institute for Artificial Intelligence [★]Apple, Inc.
[♠]Paul G. Allen School of Computer Science & Engineering, University of Washington
{michael.hassid,daniel.rotem,roy.schwartz1}@mail.huji.ac.il
haop@allenai.org {jkasai,nasmith}@cs.washington.edu ivamon@apple.com

Abstract

The attention mechanism is considered the backbone of the widely-used Transformer architecture. It contextualizes the input by computing input-specific attention matrices. We find that this mechanism, while powerful and elegant, is not as important as typically thought for pretrained language models. We introduce PAPA,[1] a new probing method that replaces the input-dependent attention matrices with constant ones—the average attention weights over multiple inputs. We use PAPA to analyze several established pretrained Transformers on six downstream tasks. We find that *without* any input-dependent attention, all models achieve competitive performance—an average relative drop of only 8% from the probing baseline. Further, little or no performance drop is observed when replacing half of the input-dependent attention matrices with constant (input-*independent*) ones. Interestingly, we show that better-performing models lose more from applying our method than weaker models, suggesting that the utilization of the input-dependent attention mechanism might be a factor in their success. Our results motivate research on simpler alternatives to input-dependent attention, as well as on methods for better utilization of this mechanism in the Transformer architecture.

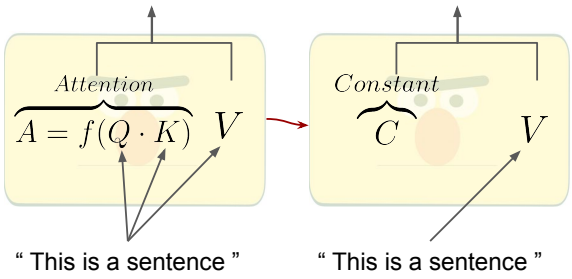


Figure 1: Illustration of the PAPA method, which measures how much PLMs use the attention mechanism. PAPA replaces the input-dependent attention matrices (left) with constant ones (right). We then measure the performance gap between the two. Moderate drop indicates minor reliance on the attention mechanism.

1 Introduction

Pretrained Transformer (Vaswani et al., 2017) models have enabled great progress in NLP in recent years (Devlin et al., 2019; Liu et al., 2019b; Raffel et al., 2020; Brown et al., 2020; Chowdhery et al., 2022). A common belief is that the backbone of the Transformer model—and pretrained language models (PLMs) in particular—is the *attention mechanism*, which applies multiple attention heads in parallel, each generating an *input-dependent* attention weight matrix.

Interestingly, recent work found that attention patterns tend to focus on constant (input-*independent*) positions (Clark et al., 2019; Voita et al., 2019), while other works showed that it is possible to pretrain language models where the attention matrices are replaced with constant matrices without major loss in performance (Liu et al., 2021; Lee-Thorp et al., 2021; Hua et al., 2022). A natural question that follows is how much standard PLMs, pretrained *with* the attention mechanism, actually rely on this input-dependent property. This paper shows that they are less dependent on it than previously thought.

We present a new analysis method for PLMs: Probing Analysis for PLMs' Attention (PAPA). For each attention head h, PAPA replaces the attention matrix with a constant one: a simple average of the attention matrices for h computed on some unlabeled corpus. Replacing all attention matrices with such constant matrices results in an attention-free variant of the original PLM (See Fig. 1). We then compute, for some downstream tasks, the probing performance gap between an original model and its attention-free variant. This provides a tool to

*This work was done while Hao Peng and Ivan Montero were at the University of Washington.

[1]PAPA stands for ***P**robing **A**nalysis for **P**LMs' **A**ttention*.

quantify the models' reliance on attention. Intuitively, a larger performance drop indicates that the model relies more on the input-dependent attention mechanism.

We use PAPA to study three established pretrained Transformers: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), and DeBERTa (He et al., 2021), each with BASE- and LARGE-sized versions. We evaluate these models on six diverse benchmarks, spanning text classification and structured prediction tasks.

Our results suggest that attention is not as important to pretrained Transformers as previously thought. First, the performance of the attention-free variants is comparable to original models: an average *relative* drop of only 8%. Second, replacing half of the attention matrices with constant ones has little effect on performance, and in some cases may even lead to performance improvements. Interestingly, our results hint that better models use their attention capability more than weaker ones; when comparing the effect of PAPA on different models, we find that the better the model's original performance is, the more it suffers from replacing the attention matrices with constant ones. This suggests a potential explanation for the source of the empirical superiority of some models over others—they make better use of the attention mechanism.

This work grants a better understanding of the attention mechanism in pretrained Transformers. It also motivates further research on simpler or more efficient Transformer models, either for pretraining (Lee-Thorp et al., 2021; Liu et al., 2021; Hua et al., 2022) or potentially as an adaptation of existing pretrained models (Peng et al., 2020a, 2022; Kasai et al., 2021). It also provides a potential path to improve the Transformer architecture—by designing inductive bias mechanisms for better utilization of attention (Peng et al., 2020b; Wang et al., 2022).

Finally, our work may contribute to the "attention as explanation" debate (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019; Bibal et al., 2022). By showing that some PLMs can perform reasonably well with constant matrices, we suggest that explanations arising from the attention matrices might not be crucial for models' success.

We summarize our main contributions. (1) We present a novel probing method—PAPA—which quantifies the reliance of a given PLM on its attention mechanism by "disabling" that mechanism for this PLM. (2) We apply PAPA to six leading PLMs, and find that our manipulation leads to modest performance drops on average, which hints that attention might not be as important as thought. (3) We show that better-performing PLMs tend to suffer more from our manipulation, which suggests that the input-dependent attention is a factor in their success. (4) Finally, we release our code and experimental results.[2]

2 Background: Attention in Transformers

Transformers consist of interleaving attention and feed-forward layers. In this work, we focus on Transformer encoder models, such as BERT, which are commonly used in many NLP applications.

The (multi headed) self-attention module takes as input a matrix $X \in \mathbb{R}^{n \times d}$ and produces a matrix $X^{out} \in \mathbb{R}^{n \times d}$, where n denotes the number of input tokens, each represented as a d-dimensional vector. Each attention layer consists of H heads, and each head $h \in \{1, \ldots, H\}$ has three learnable matrices: $W_Q^h, W_K^h, W_V^h \in \mathbb{R}^{d \times d'}$.[3] Multiplying them with the input X results in: $Q^h, K^h, V^h \in \mathbb{R}^{n \times d'}$ (the queries, keys and values, respectively).

The queries and the keys compute a $n \times n$ attention weight matrix A^h between all pairs of tokens as softmax-normalized dot products:[4]

$$A^h = \text{softmax}\Big(\underbrace{(X \cdot W_Q^h)}_{Q^h} \cdot \underbrace{(X \cdot W_K^h)}_{K^h}^\top\Big) \in \mathbb{R}^{n \times n} \quad (1)$$

where the softmax operation is taken row-wise. The value matrix V^h is then left-multiplied by the attention matrix A^h to generate the attention head output.

Importantly, the attention matrix A^h is *input-dependent*, i.e., defined by the input X. This property is considered to be the backbone of the attention mechanism (Vaswani et al., 2017).

An intriguing question is the extent to which PLMs actually rely on the attention mechanism. In the following, we study this question by replacing the attention matrices of PLMs with constant matrices. We hypothesize that if models make heavy use of attention, we will see a large drop in performance when preventing the model from using it. As shown below, such performance drop is often *not* observed.

[2] https://github.com/schwartz-lab-NLP/papa

[3] d' is the head-dimension, and usually defined as $d' = \frac{d}{H}$.

[4] Some attention variants (e.g., He et al., 2021) incorporate positional information as part of the calculation of A^h.

3 The PAPA Method

We present PAPA, a probing method for quantifying the extent to which pretrained Transformer models use the attention mechanism. PAPA works by replacing the Transformer attention weights with constant matrices, computed by averaging the values of the attention matrices over unlabeled inputs (Sec. 3.1). PAPA also allows for replacing any subset (not just all) of the attention matrices. We propose a method for selecting which heads to replace (Sec. 3.2). The resulting model is then probed against different downstream tasks (Sec. 3.3). The performance difference between the original and the new models can be seen as an indication of how much the model uses its attention mechanism.

3.1 Generating Constant Matrices

To estimate how much a pretrained Transformer m uses the attention mechanism, we replace its attention matrices with a set of constant ones, one for each head. To do so, PAPA constructs, for a given head h,[5] a constant matrix C^h by averaging the attention matrix A^h over a corpus of raw text. More specifically, given a corpus $D = \{e_1, \ldots, e_{|D|}\}$, C^h is defined as:

$$C^h = \frac{1}{|D|} \sum_{i=1}^{|D|} A_i^h, \tag{2}$$

where A_i^h is the input-dependent attention matrix that h constructs while processing e_i. We note that the average is taken entry-wise, and only over non-padded entries (padding tokens are ingored).

We emphasize that the construction process of C^h matrices requires no labels. In Sec. 5.2 we compare our method of constructing constant matrices from unlabeled data to other alternatives that either use no data at all, or use labeled data.

3.2 Replacing a Subset of the Heads

Different attention heads may have different levels of dependence on attention. We therefore study the effect of replacing a subset of the heads, and keeping the rest intact. To do so, we would like to estimate the reliance of each head on the input-dependent attention, which would allow replacing only the heads that are least input-dependent for the model.

[5] We do so for all layers in parallel. Layer indices omitted for simplicity.

To estimate this dependence, we introduce a new weighting parameter $\lambda^h \in (0, 1)$, initialized as $\lambda^h = 0.5$, for each attention head h.[6] λ^h is a learned weighting of the two matrices: the attention matrix A^h and the constant matrix C^h from (1) and (2) respectively. For each input e_i, a new matrix B^h is constructed as:

$$B_i^h = \lambda^h \cdot A_i^h + (1 - \lambda^h) \cdot C^h \tag{3}$$

We interpret a smaller λ^h as an indication of h less depending on the attention mechanism.

We then train the probing classifier (Sec. 3.3) along with the additional λ^h parameters. We use the learned λ^hs to decide which heads should be replaced with constant matrices, by only replacing the $k\%$ attention heads with the smallest λ^h values for some hyperparameter k.[7] Importantly, this procedure is only used as a pre-processing step; our experiments are trained and evaluated without it, where $k\%$ of each model's heads are replaced, and $(1 - k\%)$ remain unchanged.

3.3 Probing

Our goal is to evaluate how much attention a given PLM uses. Therefore, we want to avoid finetuning it for a specific downstream task, as this would lead to changing all of its weights, and arguably answer a different question (e.g., how much attention does a *task-finetuned* PLM use). Instead, we use a probing approach (Liu et al., 2019a; Belinkov, 2022) by freezing the model and adding a classifier on top.

Our classifier calculates for each layer a weighted (learned, non-attentive) representation of the different token representations. It then concatenates the different layer weighted representations, and applies a 2-layer MLP. For structured prediction tasks (e.g., NER and POS), where a representation for each token is needed, we concatenate for each token the representations across layers, and apply a 2-layer MLP.

When PAPA is applied to some input, we replace the attention matrices A^h with the corresponding constant matrices C^h.[8] We then compare the downstream performance of the original model m with the new model m'. The larger the performance gap between m and m', the higher m's dependence on the attention mechanism.

[6] λ^h is the output of a sigmoid over a learned parameter.

[7] In Sec. 5.4 we show that this head selection method outperforms other alternatives.

[8] To minimize model changes, we also mask the C^h entries corresponding to padded tokens, and normalize the matrix (row-wise), as in a regular Transformer.

3.4 Method Discussion

Contextualization with PAPA PAPA replaces the attention matrices with constant ones, which results in an attention-free model. Importantly, unlike a feed-forward network, the representations computed via the resulting model are still contextualized, i.e., the representation of each word depends on the representations of all other words. The key difference between the standard Transformer model and our attention-free model is that in the former the contextualization varies by the input, and for the latter it remains fixed for all inputs.

Potential Computational Gains The replacement of the attention matrix with a constant one motivates the search for efficient attention alternatives. Using constant matrices is indeed more efficient, reducing the attention head time complexity from $2n^2d' + 3nd'^2$ to $n^2d' + nd'^2$,[9] which shows potential for efficiency improvement.

Several works used various approaches for replacing the attention mechanism with constant ones during the pretraining phase (Lee-Thorp et al., 2021; Liu et al., 2021; Hua et al., 2022), and indeed some of them showed high computational gains. Our work tackles a different question—how much do PLMs, which trained with the attention mechanism, actually use it. Thus, unlike the approaches above, we choose to make minimal changes to the original models. Nonetheless, our results further motivate the search for efficient attention variants.

4 Experiments

We now turn to use PAPA to study the attention usage of various PLMs.

4.1 Experimental Setup

Our experiments are conducted over both text classification and structured prediction tasks, all in English. For the former we use four diverse benchmarks from the GLUE benchmark (Wang et al., 2019): MNLI (Williams et al., 2018), SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), and CoLA (Warstadt et al., 2019). For the latter we use named entity recognition (NER) and part of speech tagging (POS) from the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003).[10] We use the standard train/validation splits, and report validation results in all cases.[11]

We use three widely-used pretrained Transformer encoder models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), and DeBERTa (He et al., 2021). We use both BASE (12 layers, 12 heads in each layer) and LARGE (24 layers, 16 heads per layer) versions. For each model and each task, we generate the constant matrices with the given (unlabeled) training set of that task. In Sec. 5.3 we show that PAPA is not very sensitive to the specific training set being used.

All experiments are done with three different random seeds, average result is reported (95% confidence intervals are shown). Pre-processing and additional experimental details are described in App. A and B, respectively.

4.2 Probing Results

The results of the BASE and LARGE models are presented in Fig. 2a and 2b, respectively. We measure the performance of each model on each task using $\{1, \frac{1}{2}, \frac{1}{8}, \frac{1}{16}, 0\}$ of the model's input-dependent attention matrices and replacing the rest with constant ones.

We first consider the original, fully-attentive, models, and find that performance decreases in the order of DeBERTa, RoBERTa, and BERT. This order is roughly maintained across tasks and model sizes, which conforms with previous results of fine-tuning these PLMs (He et al., 2021). This suggests that the model ranking of our probing method is consistent with the standard fine-tuning setup.

We note that the trends across tasks and models are similar; hence we discuss them all together in the following (up to specific exceptions).

Replacing *all* attention matrices with constant ones incurs a moderate performance drop As shown in Fig. 2, applying PAPA on all attention heads leads to an 8% relative performance drop on average and not greater than 20% from the original model.[12] This result suggests that pretrained models only moderately rely on the attention mechanism.

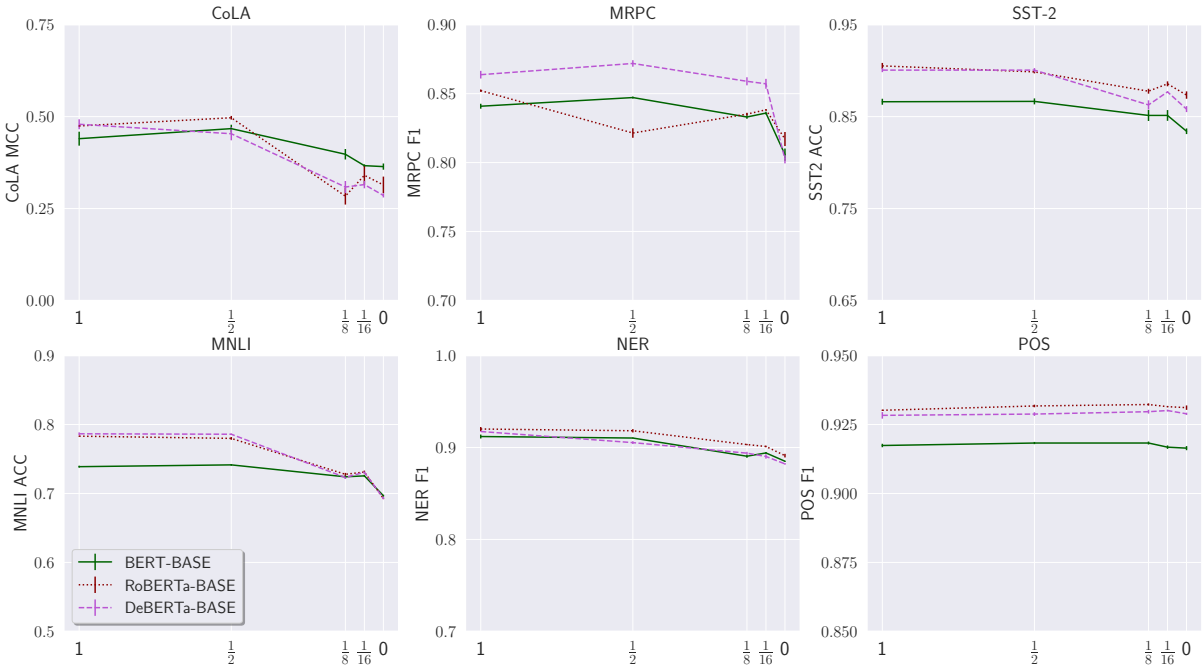
Half of the attention matrices can be replaced without loss in performance We note that in almost all cases replacing half of the models' attention matrices leads to no major drop in performance. In fact, in some cases, performance even *improves*

[9] n is the sequence length and d' is head-dimension.

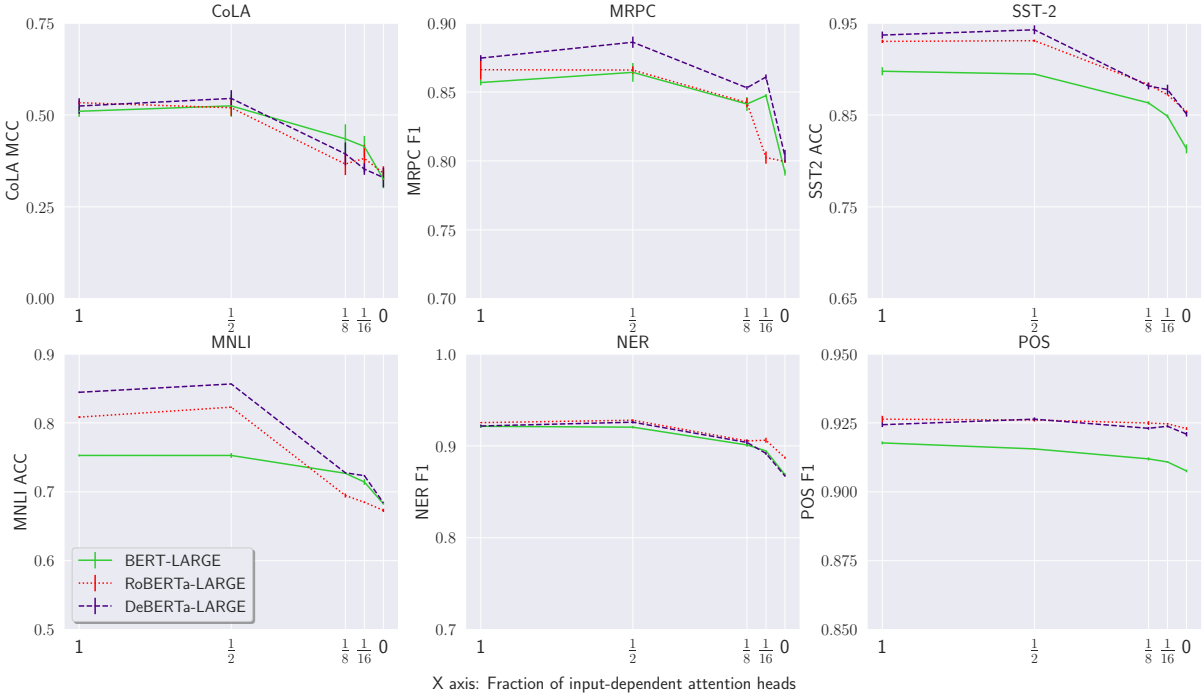
[10] We report accuracy for SST2 and MNLI, F1 score for MRPC, NER and POS, and MCC for CoLA.

[11] For MNLI, we report the mismatched validation split.

[12] For the MRPC task, some of the attention-free models do get close to the majority baseline, though still above it.



(a) BASE models



(b) LARGE models

Figure 2: Probing results (y-axis) with decreasing number of attention heads (x-axis). BASE models are shown in Fig. 2a, and LARGE models are shown in Fig. 2b. Higher is better in all cases.

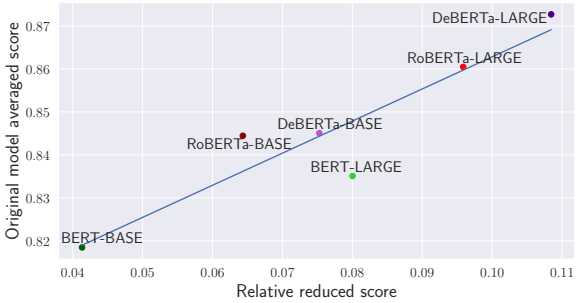


Figure 3: Stronger-performing PLMs use their attention capability more. y-axis: original model average performance; x-axis: relative reduced score when all attention matrices are replaced with constant ones.

compared to the original model (e.g., $\text{BERT}_{\text{BASE}}$ and $\text{DeBERTa}_{\text{LARGE}}$), suggesting that some of the models' heads have a slight preference towards constant matrices. This result is consistent with some of the findings of recent hybrid models that use both constant and regular attention (Liu et al., 2021; Lee-Thorp et al., 2021) to build efficient models.

Performant models rely more on attention Fig. 3 shows for each model the relation between the original performance (averaged across tasks) and the averaged (relative) reduced score when replacing all attention heads. We observe a clear trend between the models' performance and their relative reduced score, which suggests that better performing models use their attention mechanism more.

5 Further Analysis

We present an analysis of PAPA, to better understand its properties. We first discuss the patterns of the constant matrices produced by PAPA (Sec. 5.1). Next, we consider other alternatives to generating constant matrices (Sec. 5.2); we then examine whether the constant matrices are data-dependent (Sec. 5.3); we continue by exploring alternative methods for selecting which attention heads to replace (Sec. 5.4). Finally, we present MLM results, and discuss the challenges in interpreting them (Sec. 5.5). In all experiments below, we use $\text{RoBERTa}_{\text{BASE}}$. $\text{RoBERTa}_{\text{LARGE}}$ experiments show very similar trends, see App. C.

5.1 Patterns of the Constant Matrices

We first explore the attention patterns captured by different heads by observing the constant matrices (C^h). We first notice a diagonal pattern, in which each token mostly attends to itself or to its neighboring words. This pattern is observed in about 90% of the constant matrices produced by PAPA. Second, about 40% of the heads put most of their weight mass on the [CLS] and/or [SEP] tokens (perhaps in combination with the diagonal pattern described above). Lastly, while for some of the heads the weight mass is concentrated only in specific entry per row (which corresponding only to a specific token), in most of cases the weight mass is distributed over several entries (corresponding to several different tokens). These patterns are similar to those identified by Clark et al. (2019), and explain in part our findings—many of the attention heads mostly focus on fixed patterns that can also be captured by a constant matrix. Fig. 4 shows three representative attention heads that illustrate the patterns above.

5.2 Alternative Constant Matrices

PAPA replaces the attention matrices with constant ones. As described in Sec. 3.1, this procedure requires only an unlabeled corpus. In this section, we compare this choice with constant matrices that are constructed without any data (data-free matrices), and those that require labeled data for construction (labeled matrices).

For the former we consider three types of matrices: (1) Identity matrix—in which each token 'attends' only to itself, and essentially makes self-attention a regular feed-forward (each token is processed separately); (2) Toeplitz matrix—we use a simple Toeplitz matrix (as suggested in Liu et al., 2021), where the weight mass is on the current token, and it decreases as the attended token is further from the current one (the entries of the matrix are based on the harmonic series);[13] (3) Zeros matrix—essentially pruning the heads.

We also consider two types of labeled-matrices: (4) initialized as the Toeplitz matrices from (2); and (5) initialized as our average matrices. These matrices are updated during the training procedure of the probing classifier.[14]

Tab. 1 shows the performance of each attention-free resulting model for all downstream tasks. We observe that for all tasks, our average-based model

[13] Similar to the Gaussian matrices suggested by You et al. (2020).

[14] To make minimal changes to the frozen model, all constant matrices are masked and normalized (row-wise), the same as the output of the original softmax operation.

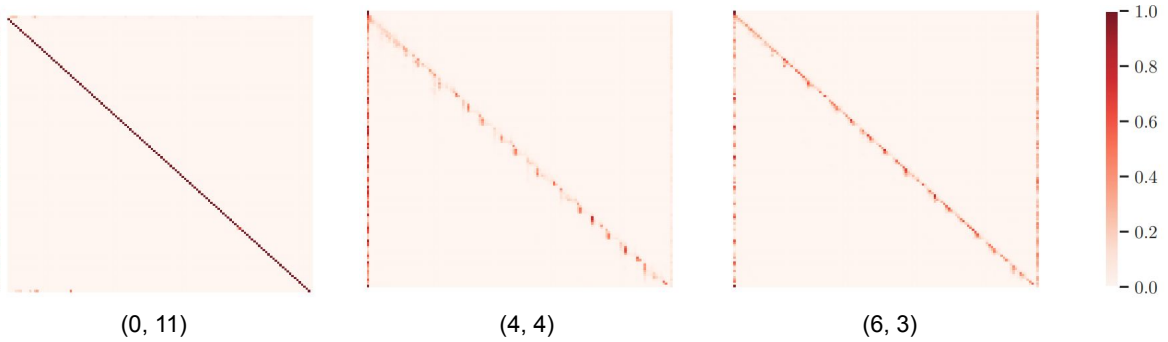


Figure 4: Generated constant matrices C^h by the PAPA method for representative heads (layer, head). These matrices used for the attention-free variant of RoBERTa$_{\text{BASE}}$ for the SST-2 task.

Matrix Construction	Matrix Type	CoLA	MRPC	SST2	MNLI-mm	NER	POS
Attention based	Original	0.47	0.85	0.91	0.78	0.92	0.93
Data-Free	Identity	0.04	0.80	0.80	0.63	0.55	0.87
	Toeplitz	0.08	0.81	0.79	0.65	0.77	0.90
	Zeros	0.09	0.80	0.80	0.66	0.57	0.87
Labeled Data	Toeplitz init.	0.08	0.81	0.79	0.68	0.78	0.91
	Average init.	**0.34**	0.81	**0.87**	**0.72**	**0.89**	**0.93**
Unlabeled Data	Average **(Ours)**	0.31	**0.82**	**0.87**	0.69	**0.89**	**0.93**

Table 1: Probe task of performance of RoBERTa$_{\text{BASE}}$ with different constant matrix types as a replacement to the input-dependent attention matrix. Bold numbers indicate the best constant model for the task. Our approach based on an average of multiple attention matrices outperforms all other data-free matrix types across all tasks, and gets similar results to the best labeled-data based model. In all tasks higher is better.

outperforms all other data-free models by a notable margin. As for the labeled-matrices models, our model also outperforms the one initialized with Toeplitz matrices (4), and in most cases gets similar results to the model initialized with average matrices (5). It should be noted that the original models (with regular attention) do not update their inner parameters in the probing training phase, which makes the comparison to the labeled-matrices models somewhat unfair. The above suggests that our choice of constant matrix replacement better estimates the performance of the attention-free PLMs.

5.3 Are the Constant Matrices Data-Dependent?

PAPA constructs the constant matrix for a given head C^h as the average of the model's attention matrices over a given corpus D, which in our experiments is set to be the training set of the task at hand (labels are not used). Here we examine the importance of this experimental choice by generating C^h using a different dataset—the MNLI training set, which is out of distribution for the other tasks.

Task	CoLA	MRPC	SST2	NER	POS
Per-Task	0.31	0.82	0.87	0.89	0.93
MNLI	0.32	0.81	0.87	0.89	0.93

Table 2: Comparison of probe task performance of RoBERTa$_{\text{BASE}}$ between two setups of constructing the averaged constant matrices C^h: Per-Task uses the task training set, while MNLI uses the constant matrices generated with the MNLI dataset. The results are similar between the two setups, which indicates a low dependence of the constant matrices on the dataset used for constructing them.

Results are presented in Tab. 2. The performance across all tasks is remarkably similar between generating the matrices using the specific task training set and MNLI, which suggests that the constant matrices might be somewhat data-independent.

5.4 Alternative Head Selection Methods

We compare our method for selecting which heads to replace (Sec. 3.2) with a few alternatives. The first two replace the heads by layer order: (1) we sort the heads from the model's first layer to the

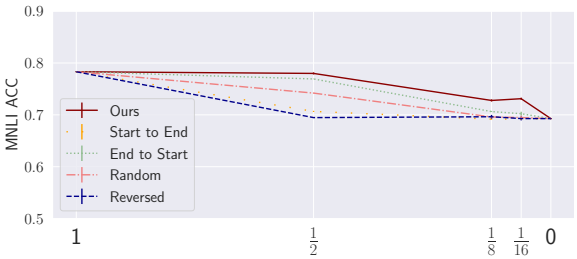


Figure 5: Comparison between different heads selection methods over MNLI. Our method outperforms all other alternatives. The x-axis represents the fraction of input-dependent attention heads.

last and (2) from the model's last layer to the first. In both cases we use the internal head ordering per layer for ordering within the layer. We then replace the first $k\%$ of the heads. We also add (3) a random baseline that randomly replaces $k\%$ of the heads, and a (4) 'Reversed' one which replaces the heads with the *highest* (rather than lowest) λ^h values (Sec. 3.2).

Fig. 5 shows the MNLI performance of each method as a function of the fraction of heads replaced. We observe that our method, which is based on learned estimation of attention importance, outperforms all other methods for every fraction of heads replaced. Moreover, the 'Reversed' method is the worst among the examined methods, which suggests that our method not only replaces the least attention dependent heads first, but also replaces the most dependent ones last. Although our head replacement order outperforms the above methods, we note that our order is an overestimation of the model attention dependency, and better methods might show that even less attention is needed.

5.5 Effects on MLM Perplexity

So far we have shown that applying PAPA on downstream tasks only incurs a moderate accuracy drop. This section aims to explore its impact on masked language modeling (MLM). We find that while our models suffer a larger performance drop on this task compared to the other tasks, this can be explained by their pretraining procedure.

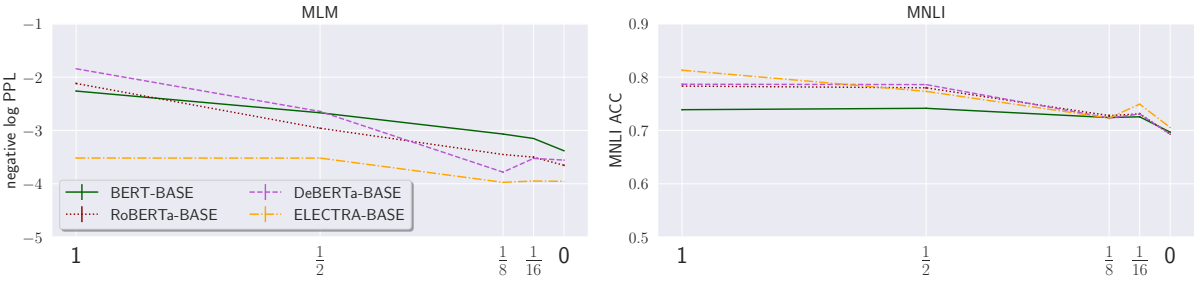
Fig. 6a plots the negative log perplexity (higher is better) of all BASE models on the WikiText-103 (Merity et al., 2017) validation set. When replacing attention matrices using PAPA, MLM suffers a larger performance drop compared to the downstream tasks (Sec. 4.2). We hypothesize that this is because these pretrained Transformers are more specialized in MLM, the task they are pretrained on. As a result, they are less able to adapt to architectural changes in MLM than in downstream tasks. To test our hypothesis, we probe $\text{ELECTRA}_{\text{BASE}}$ (Clark et al., 2020) using PAPA. ELECTRA is an established pretrained Transformer trained with the *replaced token detection* objective, instead of MLM. It has proven successful on a variety of downstream tasks.

$\text{ELECTRA}_{\text{BASE}}$'s probing performance on MLM supports our hypothesis: We first note that its original performance is much worse compared to the other models (–3.51 compared to around –2 for the MLM-based models), despite showing similar performance on downstream tasks (Fig. 6b), which hints that this model is much less adapted to MLM. Moreover, the drop when gradually removing heads is more modest (a 0.44 drop compared to 1–1.5 for the other models), and looks more similar to $\text{ELECTRA}_{\text{BASE}}$'s probing performance on MNLI (Fig. 6b). Our results suggest a potential explanation for the fact that some pretrained Transformers suffer a larger performance drop on MLM than on downstream tasks; rather than MLM demanding higher attention use, this is likely because these models are pretrained with the MLM objective.

6 Related Work

Attention alternatives Various efforts have been made in search of a simple or efficient alternative for the attention mechanism. Some works focused on building a Transformer variant based on an efficient approximation of the attention mechanism (Kitaev et al., 2020; Wang et al., 2020; Peng et al., 2020a; Choromanski et al., 2021; Schlag et al., 2021; Qin et al., 2022). Another line of research, which is more related to our work, replaced the attention mechanism in Transformers with a constant (and efficient) one. For instance, FNet (Lee-Thorp et al., 2021) replaced the attention matrix with the Vandermonde matrix, while gMLP (Liu et al., 2021) and FLASH (Hua et al., 2022) replaced it with a learned matrix.[15] These works showed that pretraining attention-free LMs can lead to competitive performance. Our work shows that PLMs trained with attention can get competitive performance even if they are denied access to this mechanism during transfer learning.

[15] These models also added a gating mechanism, which does not change the input-independent nature of their component.



(a) MLM results

(b) MNLI mismatched results

Figure 6: $\text{ELECTRA}_{\text{BASE}}$ model compared with other BASE models on MLM and MNLI. In Fig. 6a $\text{ELECTRA}_{\text{BASE}}$ behaves similarly to its behavior on MNLI, but not to the other models, which are MLM-based. In Fig. 6b $\text{ELECTRA}_{\text{BASE}}$ behaves similar to other models. In both graphs the x-axis represents the fraction of input-dependent attention heads, and the y-axis is the score of the specific task (higher is better).

Analysis of attention patterns Some investigations of how attention patterns in Transformers work use probing techniques. Clark et al. (2019), Ravishankar et al. (2021) and Htut et al. (2019) studied the attention behavior in BERT. Unlike the above, which only focuses on the attention patterns of the PLM, our work sheds light on the dependence of PLMs on their attention mechanism.

Pruning methods In this work we replaced the attention matrix with a constant one in order to measure the importance of the input-dependent ability. Works like Michel et al. (2019) and Li et al. (2021) pruned attention heads in order to measure their importance for the task examined. These works find that for some tasks, only a small number of unpruned attention heads is sufficient, and thus relate to the question of how much attention does a PLM use. In this work we argue that replacing attention matrices with constant ones provides a more accurate answer for this question compared to pruning these matrices, and propose PAPA, a method for constructing such constant matrices.

7 Conclusion

In this work, we found that PLMs are not as dependent on their attention mechanism as previously thought. To do so, we presented PAPA—a method for analyzing the attention usage in PLMs. We applied PAPA to several widely-used PLMs and six downstream tasks. Our results show that replacing all of the attention matrices with constant ones achieves competitive performance to the original model, and that half of the attention matrices can be replaced without any loss in performance. We also show a clear relation between a PLM's aggregate performance across tasks and its degradation when replacing all attention matrices with constant ones, which hints that performant models make better use of their attention.

Our results motivate further work on novel Transformer architectures with more efficient attention mechanisms, both for pretraining and for knowledge distillation of existing PLMs. They also motivate the development of Transformer variants that improve performance by making better use of the attention mechanism.

8 Limitations

This work provides an analysis of the attention mechanism in PLMs. Our PAPA method is based on probing rather than finetuning, which is more common use to PLMs. We recognize that the attention mechanism in finetuned PLMs might act differently than the original model, but our main focus is investigating the PLM itself, rather than its finetuned version.

Our analysis method is built on replacing the attention matrices with constant ones (Sec. 3.1). We build these constant matrices by averaging the attention matrices over a given dataset. Because of this choice, our results reflect a lower bound on the results of the optimal attention-free model, and we acknowledge that there might be methods for constructing the constant matrices that would lead to even smaller gaps from the original model. A similar argument can be applied for our heads selection method (Sec. 3.2). Importantly, better methods for these sub-tasks might further reduce the gap between the original models and the attention-free ones, which will only strengthen our argument.

Finally, we note that we used the PAPA method

with six English tasks, and recognize that results might be different for other tasks and other languages.

Acknowledgments

We thank Miri Varshavsky for the great feedback and moral support. This work was supported in part by NSF-BSF grant 2020793, NSF grant 2113530, an Ulman Fellowship, a Google Fellowship, a Leibnitz Fellowship, and a research gift from Intel.

References

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2021. Rethinking attention with Performers. In *Proc. of ICLR*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. arXiv:2204.02311.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proc. of BlackboxNLP*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proc. of ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proc. of IWP*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: decoding-enhanced bert with disentangled attention. In *Proc. of ICLR*.

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

Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc V. Le. 2022. Transformer quality in linear time. arXiv:2202.10447.

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

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. 2021. Finetuning pretrained transformers into RNNs. In *Proc. of EMNLP*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proc. of ICLR*.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2021. Fnet: Mixing tokens with fourier transforms. arXiv:2105.03824.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2021. Differentiable subset pruning of transformer heads. *Transactions of the Association for Computational Linguistics*, 9:1442–1459.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to MLPs. In *Advances in Neural Information Processing Systems*, volume 34, pages 9204–9215. Curran Associates, Inc.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proc. of NAACL*.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proc. of ICLR*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *NeurIPS*.

Hao Peng, Jungo Kasai, Nikolaos Pappas, Dani Yogatama, Zhaofeng Wu, Lingpeng Kong, Roy Schwartz, and Noah Smith. 2022. ABC: Attention with bounded-memory control. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7469–7483, Dublin, Ireland. Association for Computational Linguistics.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2020a. Random feature attention. In *International Conference on Learning Representations*.

Hao Peng, Roy Schwartz, Dianqi Li, and Noah A. Smith. 2020b. A mixture of h - 1 heads is better than h heads. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6566–6577, Online. Association for Computational Linguistics.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. 2022. cosformer: Rethinking softmax in attention. arXiv:2202.08791.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. 2021. Attention can reflect syntactic structure (if you let it). arXiv:2101.10927.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight memory systems. In *Proc. of ICML*.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

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 *Proc. of EMNLP*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proc. of CoNLL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NeurIPS*.

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, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of ICLR*.

Shanshan Wang, Zhumin Chen, Zhaochun Ren, Huasheng Liang, Qiang Yan, and Pengjie Ren. 2022. Paying more attention to self-attention: Improving pre-trained language models via attention guiding. arXiv:2204.02922.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. arXiv:2006.04768.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *TACL*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Weiqiu You, Simeng Sun, and Mohit Iyyer. 2020. Hard-coded gaussian attention for neural machine translation. arXiv:2005.00742.

A Pre-Processing

To make the replacement of the attention matrix with a constant one reasonable, we fix the position of the [SEP] token to always be the last token of the model's input, rather than separating the last input token from the padding tokens (i.e., it comes after the padding tokens rather than before them). For tasks with two sequences per example (e.g., MNLI), which are typically separated by an additional [SEP] token, we fix this token to always be the middle token of the sequence, followed by the second sentence. We recognize that this might lead to suboptimal usage of the input's sequence length, e.g., if one of the sentences is substantially longer than the other and particularly if it is longer than half of the sequence length, it would thus be trimmed. In our experiments this only happened in less than 0.2% of input samples for a single task (MNLI), but we recognize that this might happen more frequently in other datasets.

B Hyperparameters

All of our code was implemented with the Transformers library (Wolf et al., 2020). Hyperparameters for the probing classifier on downstream tasks are shown in Tab. 3.

	Learning Rate	Batch	Epochs	Seq. Len.
CoLA	2.00E-05	16	15	64
SST-2	1.00E-04	32	4	64
MNLI	2.00E-04	8	4	256
MRPC	2.00E-05	16	15	128
NER	1.00E-04	8	4	128
POS	5.00E-04	8	4	128
MLM	5.00E-04	8	2	128

Table 3: Probing classifier hyperparameters for downstream tasks.

C Further Analsys results for RoBERTa$_{\text{LARGE}}$

Tab. 4 and 5 show RoBERTa$_{\text{LARGE}}$'s analysis results for the experiments described in Sec. 5.2 and 5.3, respectively.

Matrix Construction	Matrix Type	CoLA	MRPC	SST2	MNLI-mm	NER	POS
Attention based	Original	0.53	0.87	0.93	0.81	0.93	0.93
Data-Free	Identity	0.09	0.72	0.80	0.65	0.56	0.87
	Toeplitz	0.11	0.79	0.80	0.65	0.74	0.89
	Zeros	0.09	0.80	0.81	0.66	0.57	0.87
Labeled Data	Toeplitz init.	0.11	0.78	0.80	0.68	0.75	0.89
	Average init.	**0.35**	**0.81**	**0.88**	**0.73**	**0.91**	**0.93**
Unlabeled Data	Average **(Ours)**	0.34	**0.81**	0.85	0.68	0.89	0.92

Table 4: Probe task of performance of RoBERTa$_{\text{LARGE}}$ with different constant matrix types as a replacement to the input-dependent attention matrix. Tab. 1 shows the results for RoBERTa$_{\text{BASE}}$.

Task	CoLA	MRPC	SST2	NER	POS
Per-Task	0.34	0.80	0.85	0.89	0.92
MNLI	0.35	0.81	0.85	0.88	0.92

Table 5: Comparison of probe task performance of RoBERTa$_{\text{LARGE}}$ between two setups of constructing the averaged constant matrices C^h: Per-Task uses the task training set, while MNLI uses the constant matrices generated with the MNLI dataset. Tab. 2 shows the results for RoBERTa$_{\text{BASE}}$.