

Incorporating Residual and Normalization Layers into Analysis of Masked Language Models

Goro Kobayashi¹ Tatsuki Kuribayashi^{1,2} Sho Yokoi^{1,3} Kentaro Inui^{1,3}

¹ Tohoku University ² Langsmith Inc. ³ RIKEN

goro.koba@dc.tohoku.ac.jp

{kuribayashi, yokoi, inui}@tohoku.ac.jp

Abstract

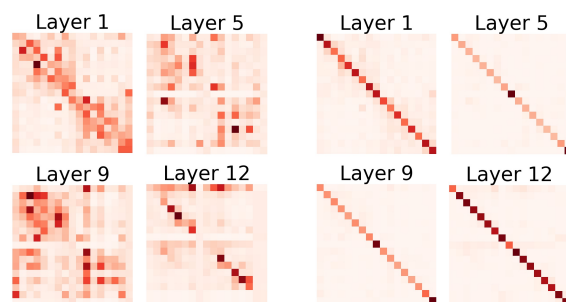
Transformer architecture has become ubiquitous in the natural language processing field. To interpret the Transformer-based models, their attention patterns have been extensively analyzed. However, the Transformer architecture is not only composed of the multi-head attention; other components can also contribute to Transformers' progressive performance. In this study, we extended the scope of the analysis of Transformers from solely the attention patterns to the whole attention block, i.e., multi-head attention, residual connection, and layer normalization. Our analysis of Transformer-based masked language models shows that the token-to-token interaction performed via attention has less impact on the intermediate representations than previously assumed. These results provide new intuitive explanations of existing reports; for example, discarding the learned attention patterns tends not to adversely affect the performance. The codes of our experiments are publicly available.¹

1 Introduction

Transformer architecture (Vaswani et al., 2017) has advanced the state of the art in a wide range of natural language processing (NLP) tasks (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020). Along with this, Transformers have become a major subject of research from the viewpoints of engineering (Rogers et al., 2020) and scientific studies (Merx and Frank, 2021; Manning et al., 2020).

In particular, the multi-head attention, a core component of Transformers, has been extensively analyzed (Clark et al., 2019; Kovaleva et al., 2019; Reif et al., 2019; Lin et al., 2019; Mareček and Rosa, 2019; Htut et al., 2019; Raganato and Tiedemann, 2018; Tang et al., 2018). While these analyses suggest that the multi-head attention contributes

¹<https://github.com/gorokoba560/norm-analysis-of-transformer>



(a) Existing analysis focusing only on the multi-head attention (Kobayashi et al., 2020).

(b) Proposed method incorporating the whole attention block (i.e., multi-head attention, residual connection, and layer normalization) into the analysis.

Figure 1: Visualizations of the token-by-token interactions in each layer when a sentence pair is fed into the pre-trained BERT-base. The diagonal elements correspond to the effect of preserving the original input information. The contrast between Figures 1a and 1b demonstrates that the contextual information contributed less to the computation of the output representations than previously expected.

to capturing linguistic information such as semantic and syntactic relations, some reports question the importance of attention. For example, several studies in fields ranging from NLP (Michel et al., 2019; Kovaleva et al., 2019) to neuroscience (Toneva and Wehbe, 2019) empirically found that discarding learned attention patterns from Transformers retains or even improves their performance in downstream tasks and the ability to simulate human brain activity. These observations imply that Transformers do not heavily rely on the multi-head attention alone, and the other components contribute to their progressive performance.

In this study, we broaden the scope of the analysis from the multi-head attention to the whole attention block, i.e., the multi-head attention, residual connection, and layer normalization. Our analysis of the Transformer-based masked language models (Devlin et al., 2019; Liu et al., 2019) revealed

that the newly incorporated components have a larger impact than expected in previous studies (Abnar and Zuidema, 2020; Kobayashi et al., 2020) (Figure 1).

More concretely, we introduce an exact decomposition of the operations in the whole attention block exploiting the norm-based analysis (Kobayashi et al., 2020). Our analysis quantifies the impact of the two contrasting effects of the attention block : (i) “mixing” the input representations via attention and (ii) “preserving” the original input mainly via residual connection (Section 3). Our analysis reveals that the preserving effect is more dominant in each attention block than previously estimated (Abnar and Zuidema, 2020; Kobayashi et al., 2020). The results also reveal the detailed mechanism of each component in the attention block. The residual connections pass through much larger vectors than the vectors produced by the multi-head attention. The layer normalization also reduces the effect of the operation via attention.

Our finding of the relatively small impact of the multi-head attention provides new intuitive interpretations for some existing reports, for example, discarding the learned attention patterns did not adversely affect their performance. Our analysis also provides a new intuitive perspective on the behaviors of Transformer-based masked language models. For example, BERT (Devlin et al., 2019) highlights low-frequency (informative) words in encoding texts, which is consistent with the existing methods for effectively computing text representations (Luhn, 1958; Arora et al., 2017).

The contributions of this study are as follows:

- We expanded the scope of Transformers analysis from the multi-head attention to the attention block (i.e., multi-head attention, residual connection, and layer normalization).
- Our analysis revealed that the operation via residual connection and layer normalization contributes more to the internal representations than expected in previous studies (Abnar and Zuidema, 2020; Kobayashi et al., 2020).
- We detailed the functioning of BERT: (i) BERT tends to mix a relatively large amount of contextual information into [MASK] in the middle and later layers; and (ii) the contribution of contextual information in the attention block is related to word frequency.

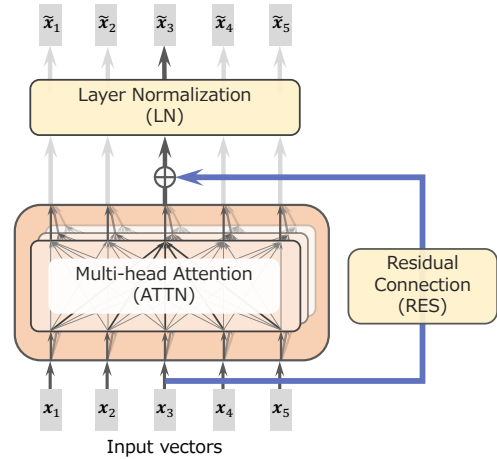


Figure 2: Visualization of the attention block, consisting of multi-head attention, residual connection, and layer normalization, in the Transformer layer.

2 Background: Transformer architecture

The Transformer architecture consists of a stack of layers. Each layer has an attention block, which is responsible for capturing the interactions between input tokens. The attention block can be further divided into the three components: **multi-head attention (ATTN)**, **residual connection (RES)**, and **layer normalization (LN)** (Figure 2). This block can be written as the following composite function:

$$\tilde{x}_i = \text{LN}\left(\text{ATTN}(x_i, \mathbf{X}) + x_i\right), \quad (1)$$

where $x_i \in \mathbb{R}^d$ is the i -th input representation, $\mathbf{X} := [x_1, \dots, x_n] \in \mathbb{R}^{n \times d}$ is the sequence of input representations, and $\tilde{x}_i \in \mathbb{R}^d$ is the output representation corresponding to x_i . Boldface letters such as \mathbf{x} denote row vectors. In the following, we review the computations in the ATTN, RES, and LN components.

Multi-head attention (ATTN): The ATTN takes the role of mixing contextual information into output representations. Formally, given input representations \mathbf{X} , the H head ATTN computes the output $\text{ATTN}(x_i, \mathbf{X}) \in \mathbb{R}^d$ for each input x_i :

$$\text{ATTN}(x_i, \mathbf{X}) = \sum_{h=1}^H \text{ATTN}^{(h)}(x_i, \mathbf{X}), \quad (2)$$

where $\text{ATTN}^{(h)}(x_i, \mathbf{X}) \in \mathbb{R}^d$ denotes the output vector from each attention head h . $\text{ATTN}^{(h)}(x_i, \mathbf{X})$ is computed by each attention

head h as follows:

$$\text{ATTN}^{(h)}(\mathbf{x}_i, \mathbf{X}) = \sum_{j=1}^n \alpha_{i,j}^{(h)} \left(\mathbf{x}_j \mathbf{W}_V^{(h)} + \mathbf{b}_V^{(h)} \right) \mathbf{W}_O^{(h)}, \quad (3)$$

$$\alpha_{i,j}^{(h)} := \text{softmax}_{\mathbf{x}_j \in \mathbf{X}} \left(\frac{\mathbf{q}(\mathbf{x}_i) \mathbf{k}(\mathbf{x}_j)^\top}{\sqrt{d_h}} \right), \quad (4)$$

$$\mathbf{q}(\mathbf{x}) := \mathbf{x} \mathbf{W}_Q^{(h)} + \mathbf{b}_Q^{(h)}, \quad (5)$$

$$\mathbf{k}(\mathbf{x}) := \mathbf{x} \mathbf{W}_K^{(h)} + \mathbf{b}_K^{(h)}, \quad (6)$$

where $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)} \in \mathbb{R}^{d \times d_h}$, and $\mathbf{W}_O^{(h)} \in \mathbb{R}^{d_h \times d}$ are the weight matrices, and $\mathbf{b}_Q^{(h)}, \mathbf{b}_K^{(h)}$, and $\mathbf{b}_V^{(h)} \in \mathbb{R}^{d_h}$ are the bias vectors. d_h denotes the dimension of each head (64 is usually used), and $d_h \times H = d$ holds. Here, Q, K , and V stand for Query, Key, and Value, respectively. Note that in a typical attention analysis, the attention weight $\alpha_{i,j}^{(h)}$ has been assumed to represent the contribution of the input \mathbf{x}_j to computing $\tilde{\mathbf{x}}_i$.

Residual connection (RES): In RES, the original input vector for the multi-head attention (\mathbf{x}_i) is added to its output vector:

$$\text{ATTN}(\mathbf{x}_i, \mathbf{X}) \mapsto \text{ATTN}(\mathbf{x}_i, \mathbf{X}) + \mathbf{x}_i. \quad (7)$$

Layer normalization (LN): LN first normalizes the input vector and then applies a transformation with learnable parameters $\gamma \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$:

$$\text{LN}(\mathbf{y}) = \frac{\mathbf{y} - m(\mathbf{y})}{s(\mathbf{y})} \odot \gamma + \beta \in \mathbb{R}^d, \quad (8)$$

where $m(\mathbf{y}) \in \mathbb{R}$ and $s(\mathbf{y}) \in \mathbb{R}$ denote the element-wise mean and standard deviation², respectively. Here, subtraction and division are also performed on an element-wise basis. The normalized vector, $(\mathbf{y} - m(\mathbf{y}))/s(\mathbf{y})$, is then transformed with γ and β ; here, \odot denotes the element-wise multiplication.

Note that analyzing the feed-forward networks in each layer is beyond the scope of this study and will be carried out as future work.

²Specifically, $m(\mathbf{y}) := \frac{1}{d} \sum_k \mathbf{y}^{(k)}$ and $s(\mathbf{y}) := \sqrt{\frac{1}{d} \sum_k (\mathbf{y}^{(k)} - m(\mathbf{y}) + \epsilon)^2}$, where $\mathbf{y}^{(k)}$ denotes the k -th element of the vector \mathbf{y} and $\epsilon \in \mathbb{R}$ is a small constant to stabilize the value.

3 Proposal: Analyzing attention blocks

For analyzing Transformers, solely observing the attention weights has been a common method (Clark et al., 2019; Kovaleva et al., 2019, etc.). We extend the scope of analysis to the whole attention block (ATTN, RES, and LN).

3.1 Strategy: Norm-based analysis

Kobayashi et al. (2020) introduced the *norm-based analysis* to extend the scope of analysis from the attention weights to the whole multi-head attention. We follow this *norm-based analysis* and extend its scope to the whole attention block.

The norm-based analysis first attempts to decompose the output vector $\tilde{\mathbf{x}}_i$ into the sum of the transformed input vectors $\{\mathbf{x}_j\}$:

$$\tilde{\mathbf{x}}_i = \sum_j F_i(\mathbf{x}_j), \quad (9)$$

where F_i is an appropriate vector-valued function. Then, the contribution of \mathbf{x}_j to $\tilde{\mathbf{x}}_i$ can be expressed by the norm of $F_i(\mathbf{x}_j)$. In the next subsection, we indicate that this norm-based method can be applied to analyzing the whole attention block. In other words, the output of the attention block is also be decomposed into the sum of the transformed input vectors without any approximation.

3.2 Decomposing output into a sum of inputs

The output $\tilde{\mathbf{x}}_i$ is decomposed into a sum of terms associated with each input \mathbf{x}_j . First, ATTN (Equation 2) can be decomposed into a sum of vectors (Kobayashi et al., 2020):

$$\text{ATTN}(\mathbf{x}_i, \mathbf{X}) = \sum_{j=1}^n \sum_{h=1}^H \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j), \quad (10)$$

$$f^{(h)}(\mathbf{x}) := \left(\mathbf{x} \mathbf{W}_V^{(h)} + \mathbf{b}_V^{(h)} \right) \mathbf{W}_O^{(h)}. \quad (11)$$

Second, in RES, no interaction between subscripts i and j occurs, and the form is already additively decomposed. Third, by exploiting the linearity of $m(\cdot)$, we can derive the ‘‘distributive law’’ of LN and decompose it. Let $\mathbf{y} = \sum_j \mathbf{y}_j$ be the input to LN. Then,

$$\text{LN}(\mathbf{y}) = \sum_j g_{\mathbf{y}}(\mathbf{y}_j) + \beta, \quad (12)$$

$$g_{\mathbf{y}}(\mathbf{y}_j) := \frac{\mathbf{y}_j - m(\mathbf{y}_j)}{s(\mathbf{y})} \odot \gamma. \quad (13)$$

See Appendix A for the derivation.

With these decompositions of ATTN and LN, the output of the whole attention block can be written as the sum of vector-valued functions with each input vector in \mathbf{X} as an argument:

$$\tilde{\mathbf{x}}_i = \text{LN}(\text{ATTN}(\mathbf{x}_i, \mathbf{X}) + \mathbf{x}_i) \quad (14)$$

$$= \text{LN}\left(\sum_{h=1}^H \sum_{j=1}^n \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j) + \mathbf{x}_i\right) \quad (15)$$

$$= \sum_{j \neq i} g_{\mathbf{y}}\left(\sum_{h=1}^H \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j)\right) + g_{\mathbf{y}}\left(\sum_{h=1}^H \alpha_{i,i}^{(h)} f^{(h)}(\mathbf{x}_i)\right) + g_{\mathbf{y}}(\mathbf{x}_i) \quad (16)$$

$$+ \beta, \\ \mathbf{y} := \text{ATTN}(\mathbf{X}, \mathbf{x}_i) + \mathbf{x}_i. \quad (17)$$

3.3 Measuring the contribution of context

Regarding the success of the contextualized representations in NLP, an interesting issue is the *location and strength of the context mixing performed in the model*. Based on this issue, we investigate the attention block by categorizing the terms in Equation 16 into the two effects:³

1. *Mixing* contextual information into the output representation by the ATTN:

$$\tilde{\mathbf{x}}_{i \leftarrow \text{context}} := \sum_{j \neq i} g_{\mathbf{y}}\left(\sum_{h=1}^H \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j)\right).$$

We measure the magnitude of this *context-mixing effect* by the norm $\|\tilde{\mathbf{x}}_{i \leftarrow \text{context}}\|$. This strength refers to the amount of information from the surrounding contexts $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \setminus \{\mathbf{x}_i\}$ in calculating $\tilde{\mathbf{x}}_i$.

2. *Preserving* the original information via ATTN and RES:

$$\tilde{\mathbf{x}}_{i \leftarrow i} := g_{\mathbf{y}}\left(\sum_{h=1}^H \alpha_{i,i}^{(h)} f^{(h)}(\mathbf{x}_i)\right) + g_r(\mathbf{x}_i).$$

We measure the magnitude of the *preserving effect* by the norm $\|\tilde{\mathbf{x}}_{i \leftarrow i}\|$. This strength refers to the amount of information from the original vector \mathbf{x}_i used in calculating $\tilde{\mathbf{x}}_i$. At the attention block, information from the input vector \mathbf{x}_i can flow through two ways: (i) attention to the original input (the first term) and (ii) residual connection (the second term).

³The bias β affects neither the *context-mixing* effect nor the *preserving* effect. Thus, we ignored this term in our analysis.

To summarize the relative strength of the context-mixing effect, the **context-mixing ratio** is defined as follows:

$$r_i = \frac{\|\tilde{\mathbf{x}}_{i \leftarrow \text{context}}\|}{\|\tilde{\mathbf{x}}_{i \leftarrow \text{context}}\| + \|\tilde{\mathbf{x}}_{i \leftarrow i}\|}. \quad (18)$$

A higher mixing ratio indicates that the mixing effect is more dominant than the preserving effect in the computation of $\tilde{\mathbf{x}}_i$.

Note that [Abnar and Zuidema \(2020\)](#) assumed that the multi-head attention and residual connection always equally impact the output, i.e., $r \approx 0.5$ in the analysis of Transformers. However, our experiments revealed that, in practical masked language models, the mixing ratio is considerably below 0.5.

4 Experiments: Analysis of mixing ratio

The context-mixing ratio of the attention blocks in pre-trained masked language models was analyzed using the proposed norm-based analysis. The obtained results were different from those of the existing methods that analyze only some of the components in the attention block.

4.1 General setup

Model: We investigated the 32 variants of the masked language models (BERT with five different sizes, BERT-base trained with 25 different seeds, and RoBERTa with two different sizes). In Section 4, the results for BERT-base and RoBERTa-base are demonstrated. The results for the other models are provided in Appendix B and C. Note that most of our findings reported in this section generalize across these model variants. Exceptions are discussed in the relevant section (Section 4.4).

Data: We experimented with the following four datasets: (i) Wikipedia ([Clark et al., 2019](#)), (ii) the Stanford Sentiment Treebank v2 dataset (SST-2, [Socher et al., 2013](#)), (iii) the Multi-Genre Natural Language Inference corpus (MNLI, [Williams et al., 2018](#)), and (iv) the standard CoNLL-2003 Named Entity Recognition dataset (CoNLL'03-NER, [Tjong Kim Sang and De Meulder, 2003](#)). The statistics of the datasets are shown in Table 1. Owing to the limitation of space, we only give the results for the Wikipedia data in Section 4. The trends observed for the Wikipedia dataset were generalized across the other datasets (see Appendix B). Note that each sequence of the Wikipedia dataset

Data	#Samples	Avg. length	Domains
Wikipedia	992	122	Web encyclopedia
SST-2	872	25	Movie reviews
MNLI	1000	39	10 distinct genres
CoNLL'03-NER	1000	21	News

Table 1: Details of the datasets. Avg. length is the number of tokens segmented by BERT per sample.

consists of paired consecutive paragraphs. Each sequence is fed into the models with masking applied to 15% of tokens 80% of the time.⁴

Analysis methods: We compared the context-mixing ratio computed with the following five analyzing methods:

- **ATTN-W:** Analyzing ATTN via attention weights, which has been applied in many existing studies (Clark et al., 2019; Kovaleva et al., 2019; Mareček and Rosa, 2019, etc.). The ratio, where attention weight assigned to the original input vector $\alpha_{i,i}$ corresponds to the preserving effect, and the others correspond to the mixing effect, is calculated as follows:

$$\frac{1}{H} \sum_{h=1}^H \frac{\sum_{j \neq i} \alpha_{i,j}^{(h)}}{\sum_{j \neq i} \alpha_{i,j}^{(h)} + \alpha_{i,i}^{(h)}} = \frac{\sum_h \sum_{j \neq i} \alpha_{i,j}^{(h)}}{H}$$

- **ATTN-N:** Analyzing ATTN via the vector norm (Kobayashi et al., 2020). The mixing ratio is calculated as

$$\frac{\|\sum_h \sum_{j \neq i} \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j)\|}{\|\sum_h \sum_{j \neq i} \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j)\| + \|\sum_h \alpha_{i,i}^{(h)} f^{(h)}(\mathbf{x}_i)\|}$$

- **ATTNRES-W:** Analyzing ATTN and RES via attention weights, as Abnar and Zuidema (2020) did. They assumed that the residual-aware attention matrix is constructed as $0.5\mathbf{A} + 0.5\mathbf{I}$. Here, \mathbf{A} is the actual attention matrix and \mathbf{I} is the identity matrix considered as the effect of residual connection. The mixing ratio is calculated as

$$\frac{1}{H} \sum_{h=1}^H \frac{\sum_{j \neq i} 0.5\alpha_{i,j}^{(h)}}{\sum_j 0.5\alpha_{i,j}^{(h)} + 0.5}$$

- **ATTNRES-N (proposed):** Analyzing ATTN and RES via the vector norm – a version of our proposed method that does not consider LN. The mixing ratio is calculated as

⁴For the other datasets, we used 1000 samples from their validation set or used all of their validation set if the number of sequences is less than 1000.

Methods	Mean	Max	Min
— BERT-base —			
ATTN-W	97.1	100.0	45.0
ATTN-N	85.2	100.0	4.9
ATTNRES-W	48.6	50.0	22.5
ATTNRES-N	22.3	65.7	2.0
ATTNRESLN-N	18.8	61.3	1.3
— RoBERTa-base —			
ATTN-W	95.8	100.0	3.8
ATTN-N	84.4	100.0	13.8
ATTNRES-W	47.9	50.0	1.9
ATTNRES-N	19.6	69.9	1.8
ATTNRESLN-N	16.2	73.4	1.5

Table 2: Mean, maximum, and minimum values of the mixing ratio computed with each method.

$$\frac{\|\sum_h \sum_{j \neq i} \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j)\|}{\|\sum_h \sum_{j \neq i} \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j)\| + \|\sum_h \alpha_{i,i}^{(h)} f^{(h)}(\mathbf{x}_i) + \mathbf{x}_i\|}$$

- **ATTNRESLN-N (proposed):** Analyzing ATTN, RES, and LN via the vector norm – the method proposed in Section 3. This corresponds to the r_i in Equation 18.

4.2 Results

We computed the mixing ratio of each token in each layer (each attention block) of the models with the five analysis methods (Section 4.1). The average, maximum, and minimum mixing ratios are shown in Table 2. Each row corresponds to a different analysis method.

Lower mixing ratio than in existing methods:

Table 2 shows that the mixing ratios obtained from the proposed ATTNRES-N and ATTNRESLN-N largely differ from those obtained from the existing methods. Whereas the attention analyses (ATTN-W and ATTN-N) yield mixing ratios of 84–97% and ATTNRES-W yields 48%–49%, our proposed method (ATTNRESLN-N) yields about 16 and 19% on average. The visualizations of the token-by-token interactions in the common attention map style become almost diagonal patterns (Figure 1). These demonstrate that each layer’s context mixing is lower than previously expected, and RES and LN largely cancel the mixing by ATTN. Observing the only ATTN and making an inference about the Transformer layer may lead to misleading. Note that Srivastava et al. (2015) reported a similar trend that stacked feed-forward networks tend to prioritize the “preserving” effect in skip connections.

Consistent trends across model sizes: Our method consistently shows the lowest mixing ratio among the compared methods for BERT and

RoBERTa models of various sizes (BERT-large, medium, small, tiny, and RoBERTa-large) (Appendix B). Interestingly, the context-mixing ratio is higher in the models with fewer layers (37% in BERT-tiny, but 15% in BERT-large).

4.3 Connections to previous studies

Our finding of a lower mixing ratio than previously thought provides explanations for previous results and is consistent with the pre-training strategy.

Token identifiability: The low context-mixing ratio is consistent with Brunner et al. (2020)’s reports on what they called “token identifiability.” They showed that input tokens can be well predicted only from the corresponding internal representations within BERT, especially in shallower layers, suggesting that context mixing is performed little by little. Our analysis results of the whole attention block were consistent with this finding.

Masked language modeling objective: Regarding the masked token prediction task⁵ during the pre-training, BERT and RoBERTa learn to conduct the following operations for a given input sequence: (i) infilling the [MASK] with plausible words, (ii) replacing the normal (non-special) tokens that might not fit their context (i.e., randomly replaced tokens) with plausible one, and (iii) reconstructing the original input tokens that might fit their context.

In our experiments and in common practical use, most tokens in the input sequence are not masked and fit their context. Thus, BERT is assumed to reconstruct the inputs for these tokens (i.e., behave as an auto-encoder). From this point of view, the superiority of the preserving effect is the intuitive behaviors of the masked language models.

Low impact of discarding learned attention patterns: Several studies have reported the low impact of discarding the learned attention patterns in Transformers. Michel et al. (2019) and Kovalova et al. (2019) reported that the attention patterns of many attention heads in Transformers can be removed or overwritten into the uniform patterns with almost no change in their performance, and this even brought about improvements in some cases. Voita et al. (2019) also reported the same

⁵For masked language modeling in BERT and RoBERTa pre-training, 15% of the tokens are randomly chosen from the input sequence, of which 80% are replaced with [MASK], 10% are replaced with random words, and 10% are kept unchanged.

phenomenon using a pruning method with additional training. In addition, Toneva and Wehbe (2019) reported that using uniform attention in early layers of BERT instead of the learned attention patterns leads to a better ability to simulate human brain activity.

Our analysis shows that most of the attention signal is reduced by the immediately following modules, RES and LN. This fact may explain the above observations that discarding the learned attention patterns of many attention heads does not cause a severe difference.

4.4 Mechanism

How is the mixing effect conducted in multi-head attention largely suppressed in the whole attention block? We discuss the mechanism role of ATTN and LN in suppressing the mixing ratio.

ATTN reduces context-mixing ratio: RES is a mechanism that equally adds together the output of ATTN and the input in a one-to-one fashion (Equation 7). Considering this, the mixing ratio in the scope of ATTN and RES is expected to be about 50%, while the mixing ratio was actually substantially below 50% (19–22% in ATTNRES-N) (Section 4.2). This suggests that the output of ATTN is much smaller than the input; in other words, ATTN seems to have the effect of largely *shrinking* inputs to compute the output. How is this achieved?

Recall that the output of ATTN is a weighted sum of the affine-transformed vector $f^{(h)}(\mathbf{x})$ using with attention weight $\alpha_{i,j}^{(h)}$ (Equation 10). We describe and empirically show that (i) the affine transformation in ATTN has the effect of shrinking the inputs, and (ii) the attention weights and affine-transformed vectors cancel each other out on specific vectors. We describe a brief idea here and provide the detailed derivation of each equation in the Appendix C.

First, under a coarse assumption, multiple affine transformations performed in the multi-head attention can be integrated into a single one:

$$f(\mathbf{x}) := \sum_{h=1}^H f^{(h)}(\mathbf{x}). \quad (19)$$

Assume that the input vector \mathbf{x} is a sample from the standard normal distribution: $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_d)$. Then we can estimate its magnitude by $\mathbb{E}\|\mathbf{x}\| \approx \sqrt{d}$ and the magnitude after affine transformation by $\mathbb{E}\|f(\mathbf{x})\| \approx \sqrt{\sum_{k=1}^d \sigma_k^2}$, where $\sigma_1, \dots, \sigma_d$ denote

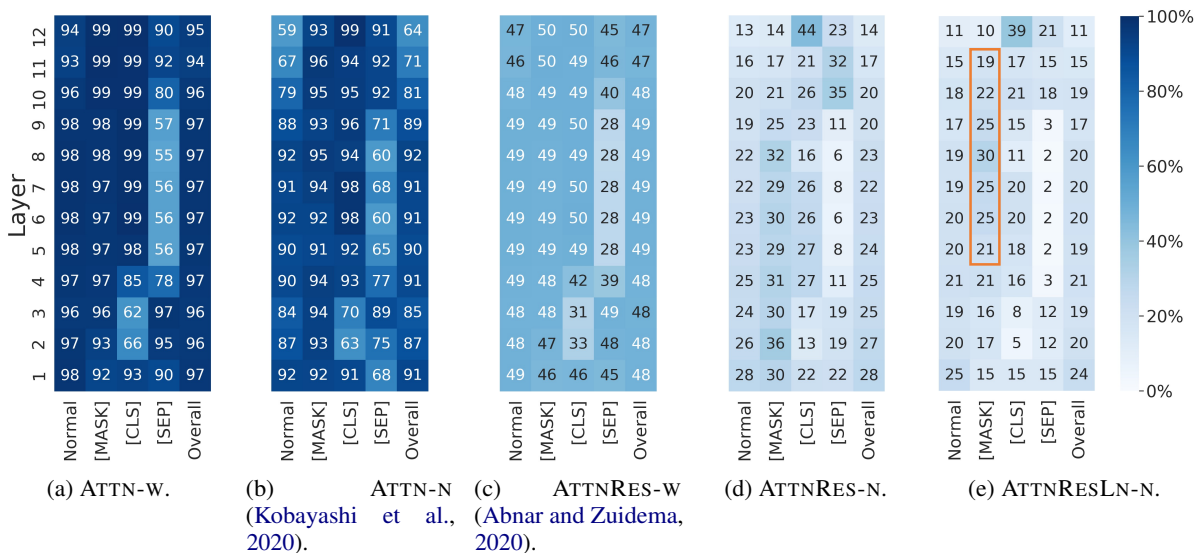


Figure 3: Mixing ratio in each layer of BERT calculated from each method.

singular values of f . Thus, the expansion rate of f is approximately estimated by

$$\|f(\mathbf{x})\|/\|\mathbf{x}\| \approx \sqrt{\sum_{k=1}^d \sigma_k^2} / \sqrt{d}. \quad (20)$$

If the ratio is lower than one, f has a tendency of shrinking the input. For commonly used large models, results stably demonstrated the shrinking tendency (layer mean of the expansion rate was $0.88 < 1.0$ for BERT-base and $0.80 < 1.0$ for BERT-large). Note that, for smaller models, results demonstrated the expanding tendency (layer mean 1.24 for BERT-mini and 1.86 for BERT-tiny). This is consistent with the result that the latter models tended to have a higher mixing ratio than the former models (Section 4.2). Detailed results are shown in Appendix C.3.

Furthermore, attention weight $\alpha_{i,j}^{(h)}$ boosts the shrinking effect. Kobayashi et al. (2020) reported the negative correlation between $\|f^{(h)}(\mathbf{x}_j)\|$ and $\alpha_{i,j}^{(h)}$ on frequent tokens. That is, ATTN wastes a lot of attention weights $\alpha_{i,j}^{(h)}$ by assigning them to small vectors $\|f^{(h)}(\mathbf{x}_j)\|$.

To summarize, ATTN’s shrinking effect is probably achieved by (i) the shrinking in f alone and (ii) further shrinking through the cancellation of α and $\|f(\mathbf{x})\|$. By these mechanisms, ATTN can contribute to decreasing the mixing ratio.

LN reduces the context-mixing ratio: LN contains not only the vector normalization but also the affine transformation with learnable parameters (Equation 8). Although the validity or usage of LN

has been investigated in terms of stability and speed of training (Parisotto et al., 2020; Liu et al., 2020), the effects of affine transformation have rarely been explored. By comparing the mixing ratios obtained from ATTBRES-N and ATTNRESLN-N (Table 2), we discovered that LN reduced the context-mixing ratio. This suggests that the scaling (by γ) of the affine transformation shrinks the vector from ATTN and emphasizes RES over ATTN.

5 Detailed analysis

We further analyzed the mixing ratio of the masked language models in detail from the perspectives of both the layer and word attributes. In this section, we inherit the experimental setup (Section 4.1) from the previous section and demonstrate results for BERT-base with the Wikipedia dataset. The results for the other experimental settings are shown in Appendix B and D. Note that only the finding reported in Section 5.2 did not generalize across model variants, and we exceptionally discuss this point in the body.

5.1 Differences by layers and token types

Figure 3 shows the mixing ratio in each layer of the BERT model (results for other models are shown in Appendix B). Each subfigure corresponds to a different analysis method, each row represents a layer, and each column represents a token type. The averaged results of the following token categories and their overall average (“overall”) are reported: (i) non-special tokens (“normal”), (ii) [MASK], (iii) [CLS], and (iv) [SEP].

Methods	Spearman’s ρ	
	all tokens	w/o special tokens
ATTN-W	0.16	0.14
ATTN-N	-0.39	-0.41
ATTNRES-W	0.16	0.14
ATTNRES-N	-0.84	-0.86
ATTNRESLN-N	-0.54	-0.58

Table 3: Spearman’s ρ between the frequency rank and the mixing ratio calculated by each method. In the “w/o special tokens” setting, it was calculated without [CLS] and [SEP].

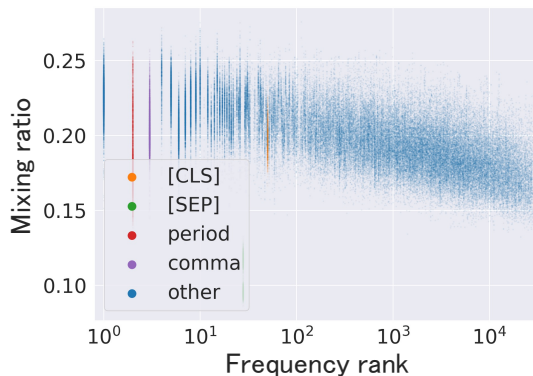


Figure 4: Relationship between the frequency rank of tokens and the mixing ratio calculated with the ATTNRESLN-N.

Results and discussion: Our proposed method showed that the mixing ratio is higher in the earlier layers than in the later ones (see the “overall” trend in Figure 3e).⁶ This trend mirrors the tendency that a deep neural network with “gates” similar to residual connections passes through the input more in the later layers (Srivastava et al., 2015).

Furthermore, our method showed a distinctive trend for the [MASK] tokens. In the middle and deep layers, the mixing ratio for [MASK] becomes higher (19–30%) than the overall trends (15–20%). Note that this trend becomes clearer when considering the RES and LN. This trend implies that in the middle and deep layers, BERT refers to contextual information for predicting the masked words. The trends of the other masked language models are shown in Appendix B.

5.2 Word frequency and mixing ratio

In this section, we will discuss the property of BERT related to the word frequency.⁷

⁶The Spearman’s ρ between the “overall” mixing ratio and the layer depth are -0.67 and -0.98 in “overall” of BERT-base and RoBERTa-base, respectively.

⁷Following Kobayashi et al. (2020), we counted the frequency for each word type by reproducing the training data of

Results: Table 3 lists the Spearman’s rank correlation ρ between the **frequency rank** (e.g., $\text{rank}(\text{“the”}) = 1$, $\text{rank}(\text{“and”}) = 6$, etc.) and the **mixing ratio** across tokens in the text data.

The results obtained from ATTNRES-N and ATTNRESLN-N indicate a surprisingly stronger negative correlation than the results obtained by the existing methods (Figure 4). This indicates that BERT discounts the information of high-frequency words compared with low-frequency ones.⁸

Discussion: Discounting high-frequency words is a common practice for making the semantic representation of a sentence or a text from word representations; examples are Luhn’s heuristic in classical text summarization (Luhn, 1958) and the smooth inverse frequency (SIF) weighting in sentence vector generation (Arora et al., 2017). Our frequency-based results reveal that attention blocks in BERT achieve this desirable property.

Our observation may also explain the phenomenon that adding up BERT’s internal or output representations does not produce a good sentence vector (Reimers and Gurevych, 2019). In contrast, in static word embeddings (e.g., word2vec (Mikolov et al., 2013)), the norm encodes the word importance derived from its frequency (Schakel and Wilson, 2015); we can generate a good sentence vector by simply adding these static word vectors (Yokoi et al., 2020). Our finding suggests that BERT encodes the token’s importance through the context-mixing ratio rather than the norm.⁹ In this sense, it is plausible that additive composition using BERT’s internal or output representations does not perform well.

Generalizability: Contrary to the other experimental results, only the relationship between word frequency and mixing ratio (Figure 4) was not consistent across different model sizes. For the larger variant (BERT-large), a stronger negative correlation between them was indicated than for BERT-base, while for the smaller variants (BERT-medium, BERT-small, BERT-mini, and BERT-tiny), even a positive correlation or no correlation was indicated (see Appendix D). Generally, larger BERT models

BERT.

⁸Kobayashi et al. (2020) reported that ATTN in BERT tends to discount frequent words when mixing contexts. We found even stronger trends after broadening the scope of analysis.

⁹In BERT, it may be difficult for the norm to encode the token importance, because the norm is fixed at each layer normalization.

(BERT-base and BERT-large) achieve better performance on downstream tasks. The different results across model sizes suggest that this desirable property can be learned when the representational power is sufficient.

6 Related work

6.1 Probing Transformers

As current neural-based models have an end-to-end, black-box nature, existing studies have adopted several strategies to interpret their inner workings (Carvalho et al., 2019; Rogers et al., 2020; Braşoveanu and Andonie, 2020). In analyzing Transformers, previous studies have mainly employed the following approaches: (i) observing the vanilla attention weights (Clark et al., 2019; Kovaleva et al., 2019; Reif et al., 2019; Lin et al., 2019; Mareček and Rosa, 2019; Htut et al., 2019; Raganato and Tiedemann, 2018; Tang et al., 2018) or the extended version (Brunner et al., 2020; Abnar and Zuidema, 2020), (ii) computing the gradient (Clark et al., 2019; Brunner et al., 2020), and (iii) analyzing the vector norm (Kobayashi et al., 2020). We adopted the norm-based analysis because this method can be naturally extended to the analysis of the whole attention block and it has some advantages (Kobayashi et al., 2020) that will also be discussed in the following paragraph.

As for broadening the scope of the analysis, Abnar and Zuidema (2020) modified the attention matrix to incorporate the residual connections into the analysis. However, they assumed that the multi-head attention and residual connection equally contributed to the computation of the output representations without any justification (Section 4.1). Brunner et al. (2020) employed a gradient-based approach for analyzing the interaction of input representations; however, the gradient ignores the impact of the input vector (i.e., only observing $\partial\tilde{x}_i/\partial x_j$ neglects the impact of x_j itself) as described in Section 6.2 of Kobayashi et al. (2020). Note that our norm-based analysis can include the magnitude of the impact of inputs in the analysis.

6.2 RES and LN in Transformers

Although residual connections (RES) (He et al., 2016) and layer normalization (LN) (Ba et al., 2016) have rarely been considered in probing studies, they are known to play important roles in both model performance and training convergence (Parisotto et al., 2020; Liu et al., 2020).

Dong et al. (2021) revealed that the residual connections are important in attention-based architectures. They demonstrated that the output of self-attention networks without residual connections converges to a rank-1 matrix quickly with increasing its layer depth. In addition, as a similar component to RES, Srivastava et al. (2015) proposed “gates” that adjust the amount of routing of the input information. Their experiments using stacked feed-forward networks for image classification also show consistent trends with ours – the effect of preserving the original input is dominant especially in the later layers. Inspired by this observation, Liu et al. (2020) modified the Transformer architecture to enhance the original input in the residual connections and demonstrated that this extension leads to better performance and convergence. Note that several variants of the Transformer-based architecture with different arrangements of RES and LN have also been proposed (Klein et al., 2018; Xiong et al., 2020; Parisotto et al., 2020), and analyzing these models is one of our future works.

7 Conclusions

In this paper, we have extended a norm-based analysis to broaden the scope of analyzing Transformers from the multi-head attention alone to the whole attention block, i.e., multi-head attention, residual connection, and layer normalization. Our analysis of the masked language models revealed that the context-mixing ratio in each block is much lower than expected in previous studies, demonstrating that RES and LN largely cancel the mixing by ATTN. This observation can provide new explanations for some unexpected results were reported on Transformers in fields ranging from NLP to neuroscience (e.g., discarding the learned attention patterns did not adversely affect the performance). Our detailed analysis further suggested that BERT discounts highly frequent, low-informative tokens.

Although our method is applicable to analyzing other variants of Transformers, our experiments were limited to the Transformer-based masked language models. In addition, the Transformer is not composed of only the attention block; feed-forward and embedding layers also exist. We plan to extend this work in both directions.

Acknowledgements

We would like to thank the members of the Tohoku NLP Lab for their insightful comments, particularly Benjamin Heinzerling for his valuable suggestions on content and wording. This work was supported by JST CREST Grant Number JP-MJCR20D2, Japan; JST ACT-X Grant Number JPMJAX200S, Japan; and JSPS KAKENHI Grant Number JP20J22697.

Ethical considerations

One recent issue in the whole NLP community is that neural-network-based models have non-intended *biases* (e.g., gender bias) induced during the training process. This paper gives a method for interpreting the inner workings of real-world machine learning models, which may help us understand such biased behaviors of the models in the future.

References

- Samira Abnar and Willem Zuidema. 2020. [Quantifying attention flow in transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4190–4197.
- Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. [A simple but tough-to-beat baseline for sentence embeddings](#). In *5th International Conference on Learning Representations (ICLR)*.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. [Layer Normalization](#). *arXiv preprint arXiv:1607.06450*.
- Adrian M. P. Braşoveanu and Răzvan Andonie. 2020. [Visualizing transformers for nlp: A brief survey](#). In *24th International Conference Information Visualization (IV)*, pages 270–279.
- Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. [On Identifiability in Transformers](#). In *8th International Conference on Learning Representations (ICLR)*.
- Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. [Machine Learning Interpretability: A Survey on Methods and Metrics](#). *Electronics*, 8(8), 832.
- 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.
- 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 (NAACL-HLT)*, pages 4171–4186.
- Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. [Attention is not all you need: pure attention loses rank doubly exponentially with depth](#). In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, PMLR, volume 139, pages 2793–2803.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). 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 55–65.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. [Deep residual learning for image recognition](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- 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*.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M Rush. 2018. [Opennmt: Neural machine translation toolkit](#). In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 177–184.
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. [Attention is not only a weight: Analyzing transformers with vector norms](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075.
- 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 4364–4373.
- 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 *8th International Conference on Learning Representations (ICLR)*.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. [Open Sesame: Getting Inside BERT’s Linguistic](#)

- Knowledge.** *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253.
- Fenglin Liu, Xuancheng Ren, Zhiyuan Zhang, Xu Sun, and Yuexian Zou. 2020. **Rethinking skip connection with layer normalization.** In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 3586–3598.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **RoBERTa: A Robustly Optimized BERT Pretraining Approach.** *arXiv preprint arXiv:1907.11692*.
- H. P. Luhn. 1958. **The automatic creation of literature abstracts.** *IBM Journal of Research and Development*, 2(2):159–165.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. **Emergent linguistic structure in artificial neural networks trained by self-supervision.** *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.
- David Mareček and Rudolf Rosa. 2019. **From Balustrades to Pierre Vinken: Looking for Syntax in Transformer Self-Attentions.** In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275.
- Danny Merckx and Stefan L. Frank. 2021. **Human sentence processing: Recurrence or attention?** In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 12–22.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. **Are Sixteen Heads Really Better than One?** In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 14014–14024.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. **Efficient estimation of word representations in vector space.** In *Workshop Track Proceedings of the 1st International Conference on Learning Representations (ICLR)*.
- Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew Botvinick, Nicolas Heess, and Raia Hadsell. 2020. **Stabilizing transformers for reinforcement learning.** In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, PMLR, volume 119, pages 7487–7498.
- Alessandro Raganato and Jörg Tiedemann. 2018. **An Analysis of Encoder Representations in Transformer-Based Machine Translation.** In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297.
- Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. **Visualizing and Measuring the Geometry of BERT.** In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 8594–8603.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence embeddings using Siamese BERT-networks.** 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 3982–3992.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. **A primer in BERTology: What we know about how BERT works.** *Transactions of the Association for Computational Linguistics (TACL)*, 8:842–866.
- Adriaan M. J. Schakel and Benjamin J. Wilson. 2015. **Measuring word significance using distributed representations of words.** *arXiv preprint arXiv:1508.02297*.
- Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D’Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. 2021. **The multiberts: Bert reproductions for robustness analysis.** *arXiv preprint arXiv:2106.16163, 2020*.
- 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 tree-bank.** In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. **Training very deep networks.** In *Advances in Neural Information Processing Systems 28 (NIPS)*.
- Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. **An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation.** In *Proceedings of the 3rd Conference on Machine Translation (WMT): Research Papers*, pages 26–35.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. **BERT Rediscovered the Classical NLP Pipeline.** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4593–4601.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. **Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition.** In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

- Mariya Toneva and Leila Wehbe. 2019. [Interpreting and improving natural-language processing \(in machines\) with natural language-processing \(in the brain\)](#). In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 14928–14938.
- Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Well-read students learn better: On the importance of pre-training compact models](#). *arXiv preprint arXiv:1908.08962*.
- 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 *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5998–6008.
- Roman Vershynin. 2018. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, 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 (ACL)*, pages 5797–5808.
- 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.
- Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. [On layer normalization in the transformer architecture](#). In *8th International Conference on Learning Representations (ICLR)*.
- Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. [Word rotator’s distance](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960.

A Derivation of “distributive law” of LN

In Section 3.2, we introduced the “distributive law” for LN (layer normalization) in Equations 12 and 13. Here, we show its derivation. Let $z = \sum_j z_j$ be the input to LN. Then, Equations 12 and 13 are derived as follows:

$$\text{LN}(z) = \frac{z - \frac{1}{d} \sum_k z^{(k)}}{s(z)} \odot \gamma + \beta \quad (21)$$

$$= \frac{\sum_j z_j - \frac{1}{d} \sum_k (\sum_j z_j)^{(k)}}{s(z)} \odot \gamma + \beta \quad (22)$$

$$= \sum_j \frac{z_j - \frac{1}{d} \sum_k z_j^{(k)}}{s(z)} \odot \gamma + \beta \quad (23)$$

$$= \sum_j \frac{z_j - m(z_j)}{s(z)} \odot \gamma + \beta \quad (24)$$

$$= \sum_j g_z(z_j) + \beta. \quad (25)$$

B Mixing ratio in other settings

In Sections 4 and 5, we showed the experimental results of mixing ratio for BERT-base with Wikipedia dataset. We also conducted the experiments with the pre-trained BERT-large (Devlin et al., 2019), BERT-medium, BERT-small, BERT-mini, BERT-tiny (Turc et al., 2019), and RoBERTa-large (Liu et al., 2019). Table 4 shows the architecture hyperparameters of each model. Table 5 shows the statistics of the mixing ratio for each model. Figures 5 to 11 show the mixing ratio at each layer (each attention block) of each model.

We also conducted it with the other three datasets. Table 6 shows the statistics of the mixing ratio for BERT-base on each dataset. Figures 12 to 14 show the mixing ratio at each layer of BERT-base on each dataset.

Furthermore, we conducted it with 25 BERT-base models trained with different seeds by Sellam et al. (2021). Table 7 shows the statistics of the mixing ratio for the models on the Wikipedia dataset. Figures 15 to 17 show the mixing ratio at each layer of three models (trained with 0th, 5th, 20th seeds) from them.

In Section 5.1, we showed the distinctive trend for the [MASK] tokens in BERT-base with the Wikipedia dataset. Even in the other models and with the other datasets, the mixing ratio for the masked tokens was relatively high in the middle and deep layers (Figures 5 to 14e).

Contrary to the results for the masked tokens, the trend for the beginning of sentence token ([CLS] or <s>) was different across these models (Figures 5 to 11). For BERT-large, RoBERTa-large, and RoBERTa-base, the layer with the highest mixing ratio for CLS was the first layer, while for the other models, it was the final or penultimate layer. Different trends between BERT and RoBERTa can be naturally explained by the fact that RoBERTa is pre-trained without the next sentence prediction. Although we cannot interpret the difference of trends across BERT models with various sizes, it was consistent among them in that the later layers mix contextual information into [CLS] with a relatively high mixing ratio. This implies that, in the later layers, BERT conducts some operations specialized to the next sentence prediction task. Solving such a discourse-level task in the later layers is consistent with the previous report that BERT makes lower-level decisions (e.g., part-of-speech tagging) in the earlier layers and that the later layers have high-level information (e.g., knowledge on co-reference) (Tenney et al., 2019).

C Details on the investigation of the mechanism of ATTN’s shrinking

We describe the details of Section 4.4.

C.1 Affine transformation in ATTN

Integration of each head’s affine transformation

To consider the scaling effect of the affine transformations in ATTN, we integrate each head’s affine transformation $f^{(h)}$ into one affine transformation $f : \mathbb{R}^d \mapsto \mathbb{R}^d$, under a coarse assumption. First, for simplicity, we assume that all heads in an ATTN assign the same weights

$$\alpha_{i,j}^{(1)} = \dots = \alpha_{i,j}^{(H)} \equiv \alpha_{i,j}. \quad (26)$$

Then, the computation of ATTN (Equation 10) can be rewritten as follows:

$$\text{ATTN}(\mathbf{x}_i, \mathbf{X}) = \sum_{j=1}^n \sum_{h=1}^H \alpha_{i,j}^{(h)} f^{(h)}(\mathbf{x}_j) \quad (27)$$

$$\approx \sum_{j=1}^n \alpha_{i,j} \sum_{h=1}^H f^{(h)}(\mathbf{x}_j) \quad (28)$$

$$= \sum_{j=1}^n \alpha_{i,j} f(\mathbf{x}_j), \quad (29)$$

where

$$f(\mathbf{x}) := \sum_{h=1}^H f^{(h)}(\mathbf{x}). \quad (30)$$

Concrete computation of f

From Equation 11, the affine transformation f is

$$f(\mathbf{x}) = \sum_{h=1}^H \left(\mathbf{x} \mathbf{W}_V^{(h)} + \mathbf{b}_V^{(h)} \right) \mathbf{W}_O^{(h)}. \quad (31)$$

Following the Transformer implementation, it can be further simplified as follows:

$$f(\mathbf{x}) = (\mathbf{x} \mathbf{W}_V + \mathbf{b}_V) \mathbf{W}_O, \quad (32)$$

$$\mathbf{W}_V := \begin{bmatrix} \mathbf{W}_V^{(1)} & \dots & \mathbf{W}_V^{(H)} \end{bmatrix} \in \mathbb{R}^{d \times d}, \quad (33)$$

$$\mathbf{b}_V := \begin{bmatrix} \mathbf{b}_V^{(1)} & \dots & \mathbf{b}_V^{(H)} \end{bmatrix} \in \mathbb{R}^d, \quad (34)$$

$$\mathbf{W}_O := \begin{bmatrix} \mathbf{W}_O^{(1)} \\ \vdots \\ \mathbf{W}_O^{(H)} \end{bmatrix} \in \mathbb{R}^{d \times d}. \quad (35)$$

On the difference in arguments of ATTN and f

In Section 4.4, we considered the scaling effect of ATTN, using the affine transformation f . One may wonder about the difference between arguments of ATTN (i.e., \mathbf{x}_i) and arguments of f (i.e., \mathbf{x}_j) in Equation 29. We can give two kinds of justification to this question.

In the estimation of the expansion rate, we consider the expected value. From the symmetry of \mathbf{x}_i and \mathbf{x}_j , when the expected value for \mathbf{x}_j is obtained, the expected value for \mathbf{x}_i is obtained. In the actual BERT model, it has been empirically confirmed that two token vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$ contained in the same context \mathbf{X} exist in a fairly close position ($\mathbf{x}_i \approx \mathbf{x}_j$). First, Ethayarajh (2019) found that the cosine similarity between the intra-sentence representations in BERT is much larger than 0. Second, the norm of input vectors has just been unified by the layer normalization in the previous layer. Thus, for our target models, $\mathbf{x}_i \approx \mathbf{x}_j$ is not a strong assumption.

Affine transformation as linear transformation

The affine transformation $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ in ATTN can be viewed as a linear transformation $\tilde{f} : \mathbb{R}^{d+1} \mapsto \mathbb{R}^{d+1}$. Given $\tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} & 1 \end{bmatrix} \in \mathbb{R}^{d+1}$, where 1 is concatenated to the end of each

input vector $\mathbf{x} \in \mathbb{R}^d$, the affine transformation f can be viewed as:

$$\tilde{f}(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}} \tilde{\mathbf{W}}^V \tilde{\mathbf{W}}^O \quad (36)$$

$$\tilde{\mathbf{W}}^V := \begin{bmatrix} & & 0 \\ & \mathbf{W}^V & \vdots \\ & & 0 \\ \mathbf{b}^V & & 1 \end{bmatrix} \in \mathbb{R}^{(d+1) \times (d+1)} \quad (37)$$

$$\tilde{\mathbf{W}}^O := \begin{bmatrix} & & 0 \\ & \mathbf{W}^O & \vdots \\ & & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(d+1) \times (d+1)}. \quad (38)$$

The ‘‘affine transformation’’ mentioned in Section 4.4 represent this linear transformation \tilde{f} , and we measured the singular values of \tilde{f} .

C.2 Expected expansion rate for a random vector

In the following, we introduce the derivation of the expansion rate of the affine transformation f , that is,

$$\frac{\|f(\mathbf{x})\|}{\|\mathbf{x}\|} \approx \frac{\sqrt{\sum_{k=1}^d \sigma_k^2}}{\sqrt{d}}. \quad (39)$$

We assume that the input vector \mathbf{x} is a sample from the standard normal distribution:

$$\mathbf{x} = (x_1, \dots, x_d) \sim \mathcal{N}(\mathbf{0}, I_d). \quad (40)$$

First, the expectation value of $\|\mathbf{x}\|^2$ is as follows (Vershynin, 2018):

$$\mathbb{E}_{\mathbf{x}} \|\mathbf{x}\|^2 = \mathbb{E}_{\mathbf{x}} \sum_{k=1}^d x_k^2 = \sum_{k=1}^d \mathbb{E}_{\mathbf{x}} x_k^2 = d. \quad (41)$$

Then, we have $\|\mathbf{x}\| \approx \sqrt{d}$.

Next, let the singular value decomposition of the linear transformation f is $f = U \Sigma V^\top$, where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d \times d}$ is the diagonal matrix of singular values of f . As the matrix V is orthogonal, the following random vector \mathbf{f} also follows the standard normal distribution, as does \mathbf{x} :

$$\mathbf{y} = (y_1, \dots, y_d) := V^\top \mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_d). \quad (42)$$

By the orthogonal transformation by U does not change the norm, we need to estimate $\|\Sigma\mathbf{y}\|^2$ in order to estimate $\|f(\mathbf{x})\|^2 = \|U\Sigma V^\top \mathbf{x}\|^2$.

$$\mathbb{E}_{\mathbf{y}} \|\Sigma\mathbf{y}\|^2 = \mathbb{E}_{\mathbf{y}} \sum_{k=1}^d \sigma_k^2 y_k^2 = \sum_{k=1}^d \sigma_k^2 \mathbb{E}_{\mathbf{x}} y_k^2 \quad (43)$$

$$= \sum_{k=1}^d \sigma_k^2. \quad (44)$$

Then, we have $\|f(\mathbf{x})\| \approx \sqrt{\sum_{k=1}^d \sigma_k^2}$.

To summarize,

$$\frac{\|f(\mathbf{x})\|}{\|\mathbf{x}\|} \approx \frac{\sqrt{\sum_{k=1}^d \sigma_k^2}}{\sqrt{d}}. \quad (45)$$

C.3 Results for other models

Table 8 shows the expected expansion rate of f for each model.

D Relationship between word frequency and mixing ratio in other settings

We also conducted the experiment shown in Section 5.2 with the pre-trained BERT-large, BERT-medium, BERT-small, BERT-mini, and BERT-tiny. However, we didn't do for RoBERTa-large and RoBERTa-base due to the difficulty of reproducing the pre-training dataset to count the word frequency. Table 9 lists the Spearman's rank correlation ρ between the frequency rank and the mixing ratio for each model. We discussed the inconsistency of the results across different model sizes in Section 5.2.

We also conducted it with the other three datasets. Table 10 lists the Spearman's rank correlation ρ between the frequency rank and the mixing ratio for each dataset.

Furthermore, we conducted with 25 BERT-base models trained with different seeds. Table 11 lists the Spearman's rank correlation ρ between the frequency rank and the mixing ratio for the models on the Wikipedia dataset.

Models	Hidden dim.	#Layer	#Head
BERT-large	1026	24	16
BERT-base	768	12	12
BERT-medium	512	8	8
BERT-small	512	4	8
BERT-mini	256	4	4
BERT-tiny	128	2	2
RoBERTa-large	1026	24	16
RoBERTa-base	768	12	12

Table 4: Architecture hyperparameters of each model.

Methods	Mean	Max	Min
— BERT-large —			
ATTN-W	97.4	100.0	15.0
ATTN-N	87.0	100.0	5.6
ATTNRES-W	48.7	50.0	7.5
ATTNRES-N	19.1	87.4	1.8
ATTNRESLN-N	14.9	86.6	1.6
— BERT-base —			
ATTN-W	97.1	100.0	45.0
ATTN-N	85.2	100.0	4.9
ATTNRES-W	48.6	50.0	22.5
ATTNRES-N	22.3	65.7	2.0
ATTNRESLN-N	18.8	61.3	1.3
— BERT-medium —			
ATTN-W	95.6	100.0	49.5
ATTN-N	83.4	99.9	9.7
ATTNRES-W	47.8	50.0	24.8
ATTNRES-N	20.9	49.2	3.8
ATTNRESLN-N	18.7	65.2	1.2
— BERT-small —			
ATTN-W	96.2	100.0	57.7
ATTN-N	85.3	100.0	10.3
ATTNRES-W	48.1	50.0	28.9
ATTNRES-N	29.6	80.4	6.7
ATTNRESLN-N	27.2	85.5	7.3
— BERT-mini —			
ATTN-W	95.5	100.0	50.9
ATTN-N	85.7	100.0	10.4
ATTNRES-W	47.8	50.0	25.4
ATTNRES-N	27.2	68.1	7.3
ATTNRESLN-N	26.4	70.7	6.6
— BERT-tiny —			
ATTN-W	94.1	99.9	38.6
ATTN-N	90.4	99.9	28.3
ATTNRES-W	47.1	50.0	19.3
ATTNRES-N	37.8	77.9	18.1
ATTNRESLN-N	37.3	70.4	17.6
— RoBERTa-large —			
ATTN-W	96.7	100.0	10.1
ATTN-N	87.8	99.9	15.2
ATTNRES-W	48.4	50.0	5.0
ATTNRES-N	19.8	87.8	4.3
ATTNRESLN	19.7	87.9	4.3
— RoBERTa-base —			
ATTN-W	95.8	100.0	3.8
ATTN-N	84.4	100.0	13.8
ATTNRES-W	47.9	50.0	1.9
ATTNRES-N	19.6	69.9	1.8
ATTNRESLN-N	16.2	73.4	1.5

Table 5: Mean, maximum, and minimum values of the mixing ratio in seven variants of the masked language models, computed with each method.

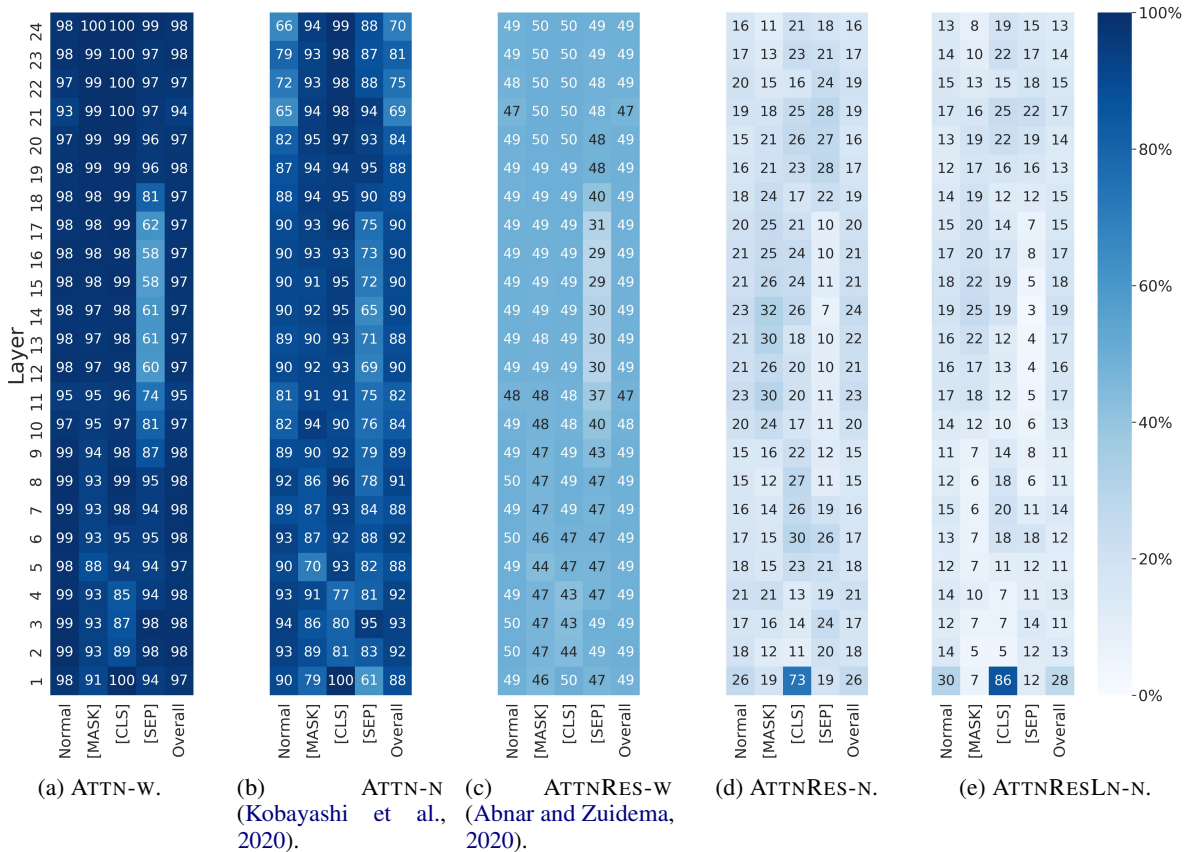


Figure 5: Mixing ratio at each layer of BERT-large calculated from each method.

Methods	Mean	Max	Min
— Wikipedia —			
ATTN-W	97.1	100.0	45.0
ATTN-N	85.2	100.0	4.9
ATTNRES-W	48.6	50.0	22.5
ATTNRES-N	22.3	65.7	2.0
ATTNRESLN-N	18.8	61.3	1.3
— SST-2 —			
ATTN-W	92.5	100.0	2.2
ATTN-N	80.3	99.8	6.7
ATTNRES-W	46.3	50.0	1.1
ATTNRES-N	22.5	50.4	2.4
ATTNRESLN-N	18.5	44.9	1.1
— MNLI —			
ATTN-W	94.6	100.0	10.0
ATTN-N	83.5	99.9	6.8
ATTNRES-W	47.3	50.0	5.0
ATTNRES-N	22.4	65.4	2.8
ATTNRESLN-N	18.3	60.7	1.2
— CoNLL'03 NER —			
ATTN-W	91.7	100.0	1.5
ATTN-N	79.0	99.9	7.0
ATTNRES-W	45.8	50.0	0.8
ATTNRES-N	22.4	51.5	2.7
ATTNRESLN-N	18.6	45.8	0.8

Table 6: Mean, maximum, and minimum values of the mixing ratio in each method for BERT-base on each data.

Methods	Mean (SD)	Max	Min
ATTN-W	96.1 (0.1)	100.0	8.8
ATTN-N	85.2 (0.4)	100.0	7.4
ATTNRES-W	48.1 (0.1)	50.0	4.4
ATTNRES-N	21.9 (0.3)	64.6	3.4
ATTNRESLN-N	17.5 (0.4)	67.7	1.4

Table 7: Mean, maximum, and minimum values of the mixing ratio in each method for 25 BERT-base models trained with different random seeds by Sellam et al. (2021). Mean value is the average of the values from 25 models, and the standard deviation (SD) is also listed. Maximum and minimum values are the maximum and minimum of these values from 25 models, respectively.

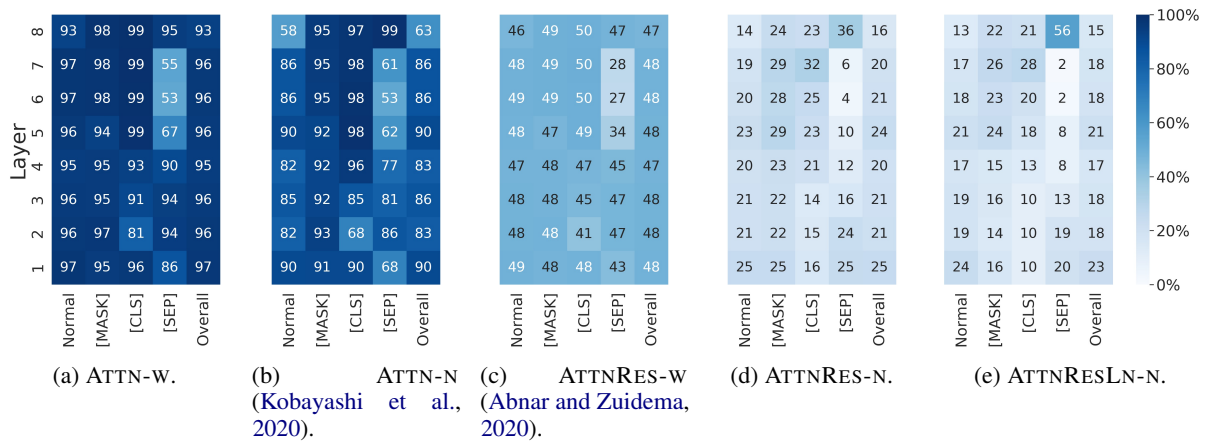


Figure 6: Mixing ratio at each layer of BERT-medium calculated from each method.

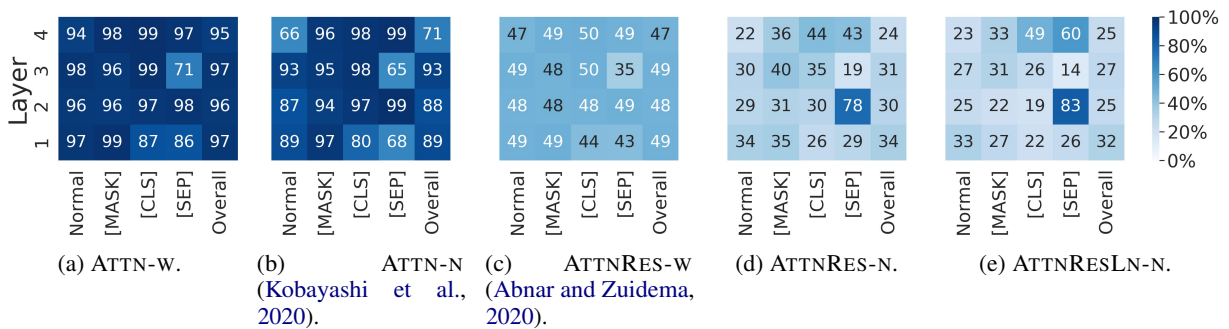


Figure 7: Mixing ratio at each layer of BERT-small calculated from each method.

Models	Mean (SD)	Min	Max
BERT-large	0.80	0.61	1.08
BERT-base	0.88	0.63	1.05
MultiBERTs (base)	0.88 (0.01)	0.65	1.43
RoBERTa-large	0.94	0.67	1.09
BERT-medium	0.87	0.60	1.41
BERT-small	1.31	0.78	2.41
BERT-mini	1.24	0.65	2.48
BERT-tiny	1.86	1.63	2.09
RoBERTa-base	1.30	1.10	1.49

Table 8: Mean, maximum, and minimum values of the scaling magnification in each layer for nine variants of the masked language models. In the “MultiBERTs (base)”, results for 25 BERT-base models trained with different random seeds by Sellam et al. (2021) are reported. Mean value is the average of the values from 25 models, and the standard deviation (SD) is also listed. Maximum and minimum values are the maximum and minimum of these values from 25 models, respectively.

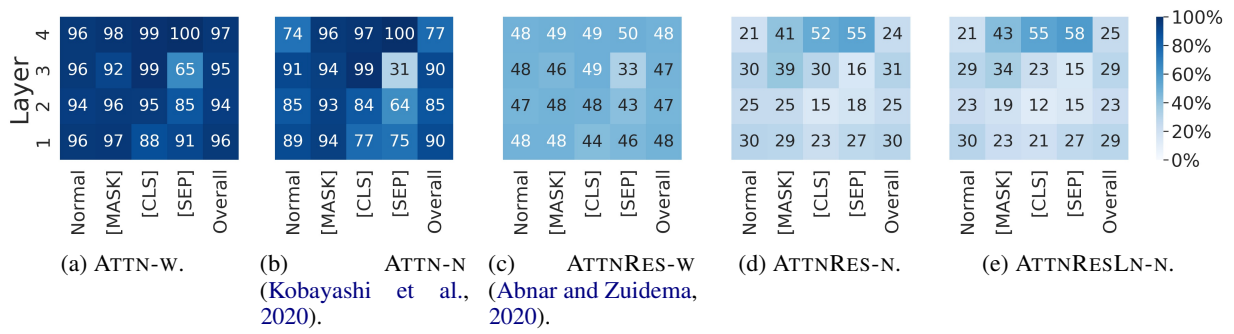


Figure 8: Mixing ratio at each layer of BERT-mini calculated from each method.

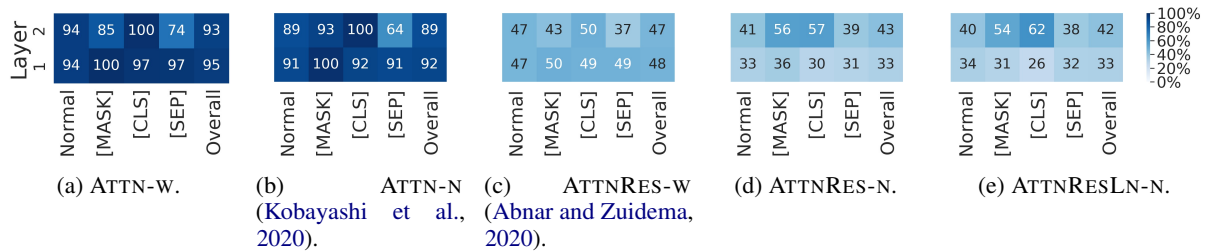


Figure 9: Mixing ratio at each layer of BERT-tiny calculated from each method.

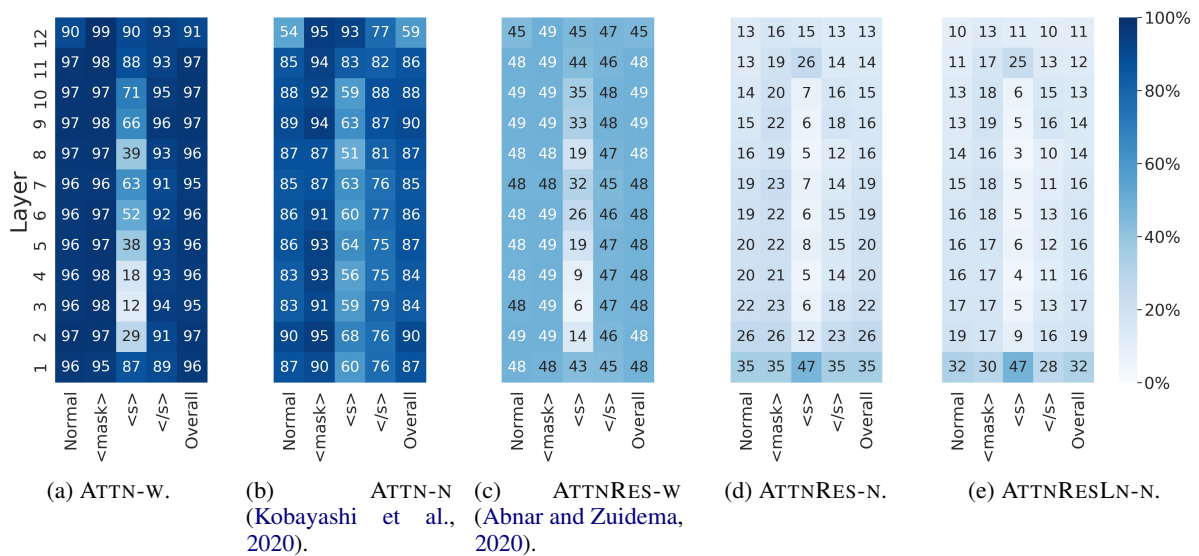


Figure 10: Mixing ratio at each layer of RoBERTa-base calculated from each method.

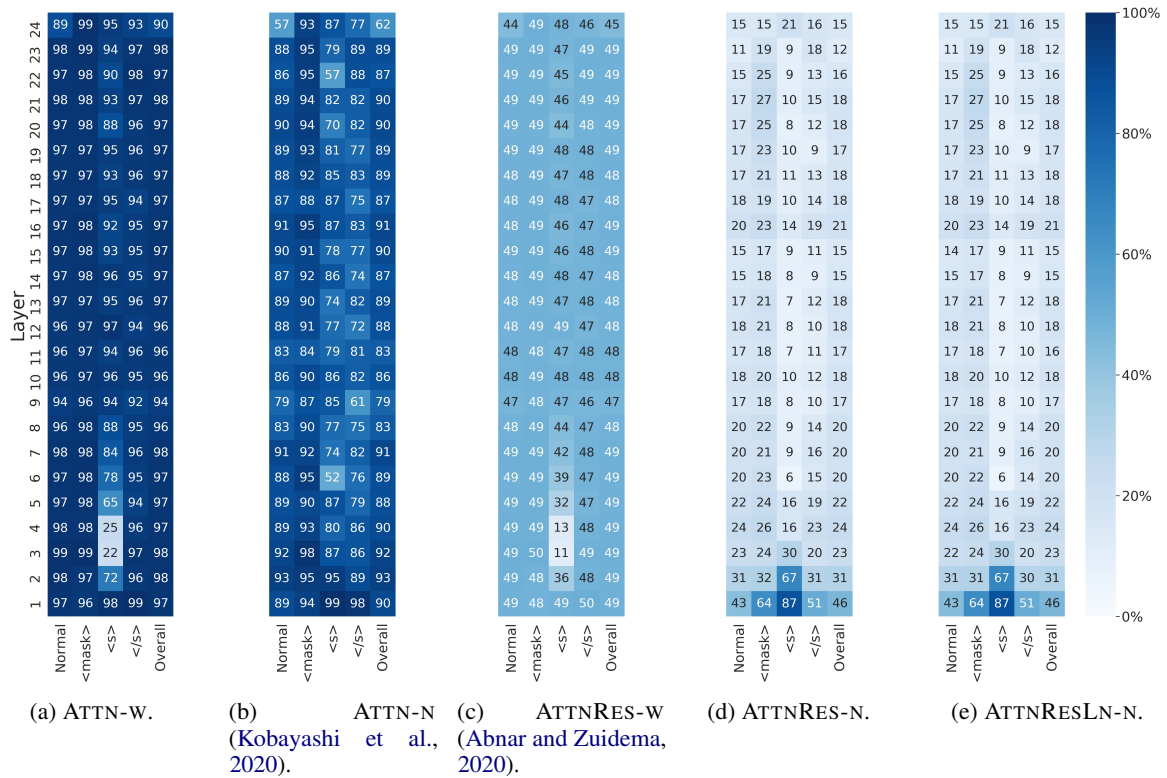


Figure 11: Mixing ratio at each layer of RoBERTa-large calculated from each method.

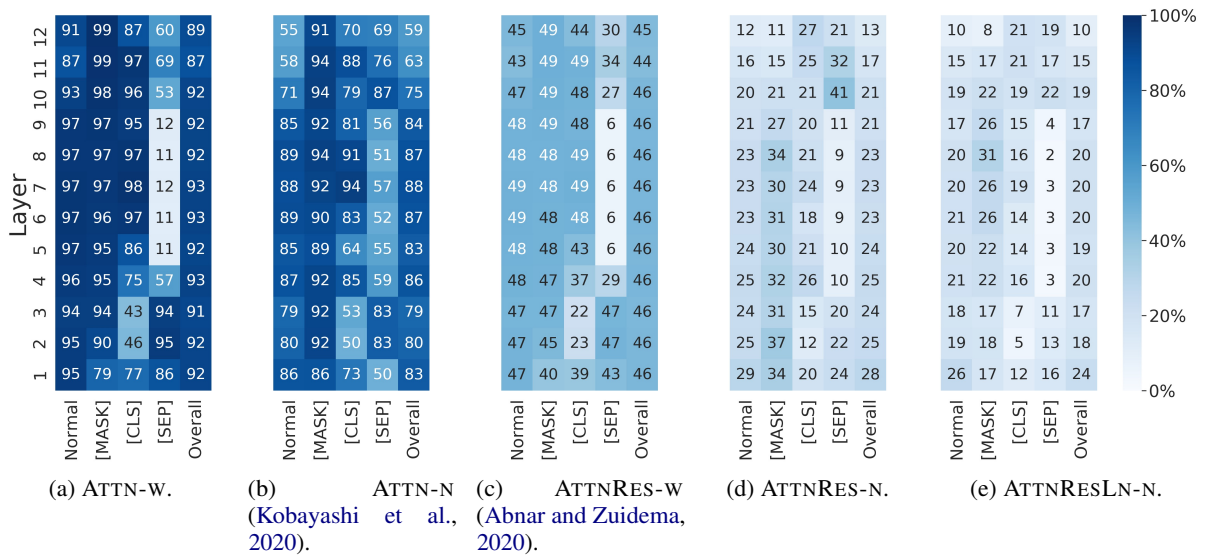


Figure 12: Mixing ratio at each layer of BERT-base calculated from each method on the SST-2.

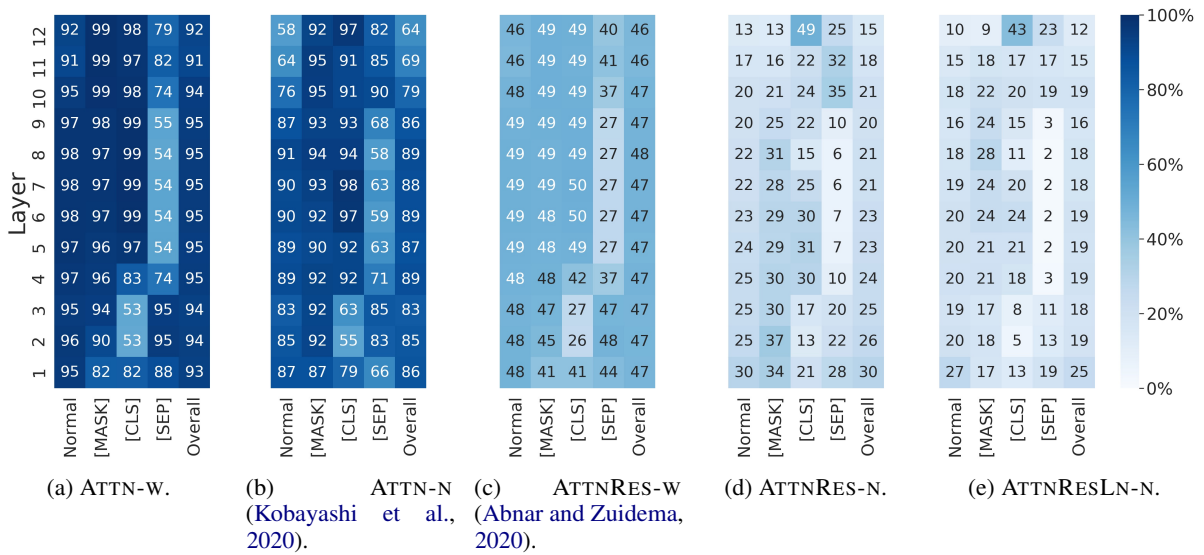


Figure 13: Mixing ratio at each layer of BERT-base calculated from each method on the MNL1.

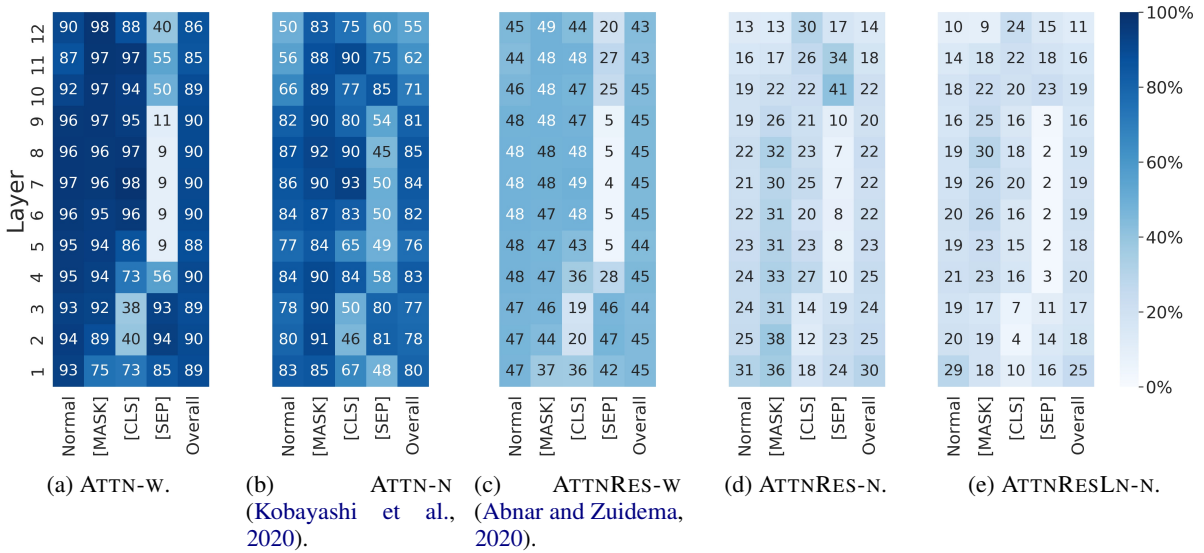


Figure 14: Mixing ratio at each layer of BERT-base calculated from each method on the CoNLL'03 NER.

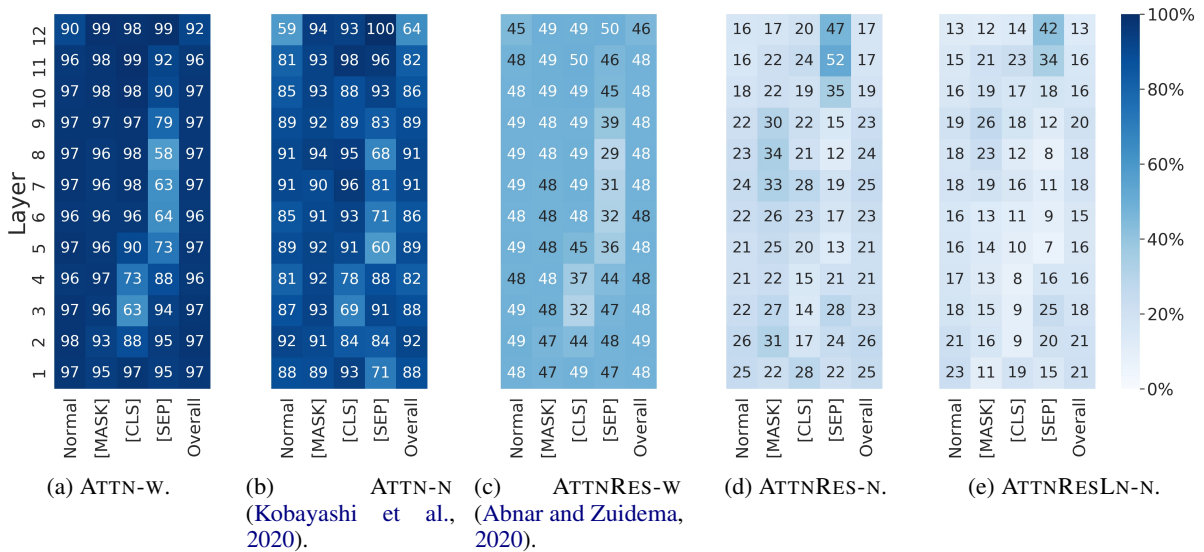


Figure 15: Mixing ratio at each layer of BERT-base trained with 0th seed.

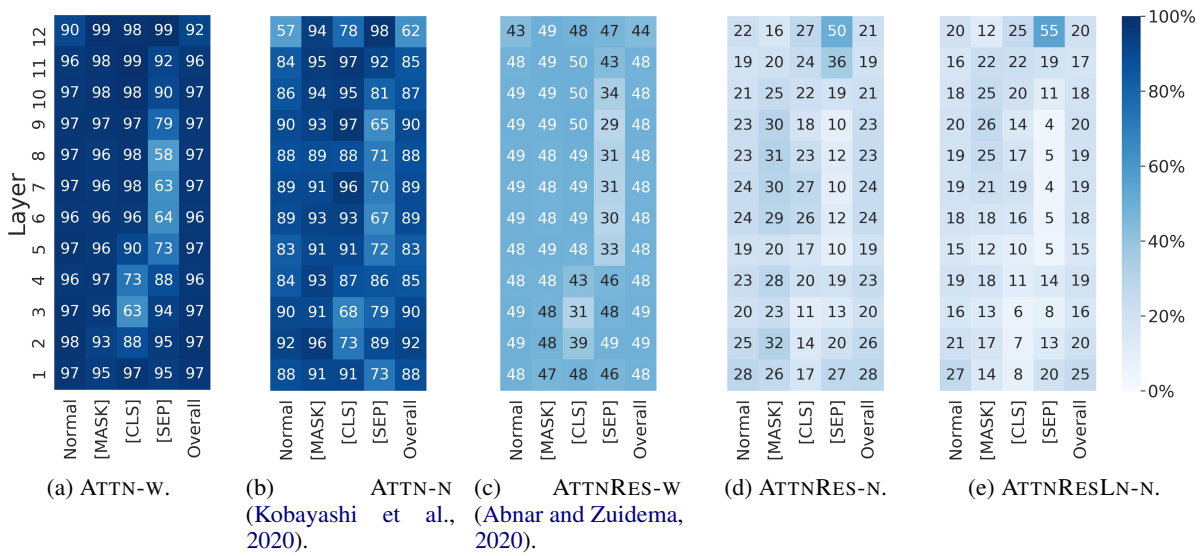


Figure 16: Mixing ratio at each layer of BERT-base trained with 10th seed.

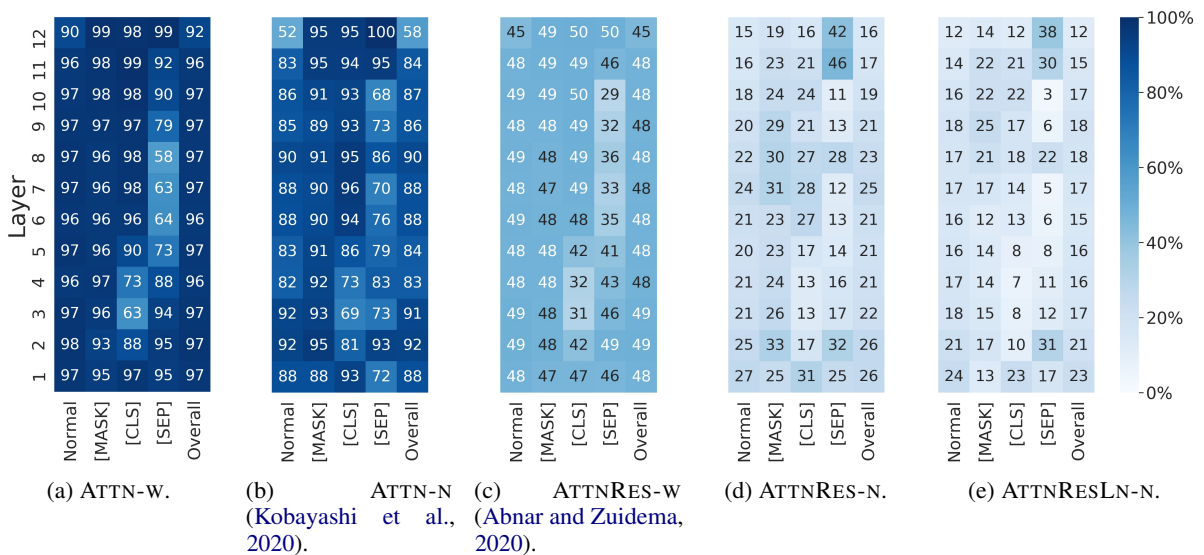


Figure 17: Mixing ratio at each layer of BERT-base trained with 20th seed.

Methods	Spearman's ρ	
	all tokens	w/o special tokens
— BERT-large —		
ATTN-W	0.44	0.44
ATTN-N	-0.53	-0.56
ATTNRES-W	0.44	0.44
ATTNRES-N	-0.83	-0.84
ATTNRESLN-N	-0.71	-0.75
— BERT-base —		
ATTN-W	0.16	0.14
ATTN-N	-0.39	-0.41
ATTNRES-W	0.16	0.14
ATTNRES-N	-0.84	-0.86
ATTNRESLN-N	-0.54	-0.58
— BERT-medium —		
ATTN-W	-0.09	-0.11
ATTN-N	-0.13	-0.14
ATTNRES-W	-0.09	-0.11
ATTNRES-N	-0.41	-0.43
ATTNRESLN-N	-0.02	-0.03
— BERT-small —		
ATTN-W	-0.05	-0.07
ATTN-N	0.26	0.26
ATTNRES-W	-0.05	-0.07
ATTNRES-N	-0.22	-0.20
ATTNRESLN-N	0.19	0.21
— BERT-mini —		
ATTN-W	-0.52	-0.55
ATTN-N	-0.15	-0.17
ATTNRES-W	-0.52	-0.55
ATTNRES-N	0.23	0.25
ATTNRESLN-N	0.42	0.44
— BERT-tiny —		
ATTN-W	-0.75	-0.77
ATTN-N	-0.62	-0.64
ATTNRES-W	-0.75	-0.77
ATTNRES-N	0.26	0.27
ATTNRESLN-N	0.24	0.25

Table 9: The Spearman’s ρ between the frequency rank and the mixing ratio calculated by each method for five variants of pre-trained BERT. In the “w/o special tokens” setting, it was calculated without [CLS] and [SEP].

Methods	Spearman's ρ	
	all tokens	w/o special tokens
— Wikipedia —		
ATTN-W	0.16	0.14
ATTN-N	-0.39	-0.41
ATTNRES-W	0.16	0.14
ATTNRES-N	-0.84	-0.86
ATTNRESLN-N	-0.54	-0.58
— SST-2 —		
ATTN-W	0.22	0.19
ATTN-N	-0.24	-0.33
ATTNRES-W	0.22	0.19
ATTNRES-N	-0.81	-0.84
ATTNRESLN-N	-0.42	-0.54
— MNLI —		
ATTN-W	0.22	0.19
ATTN-N	-0.31	-0.40
ATTNRES-W	0.22	0.19
ATTNRES-N	-0.77	-0.84
ATTNRESLN-N	-0.40	-0.50
— NER —		
ATTN-W	0.16	0.09
ATTN-N	-0.22	-0.34
ATTNRES-W	0.16	0.09
ATTNRES-N	-0.79	-0.85
ATTNRESLN-N	-0.41	-0.57

Table 10: The Spearman’s ρ between the frequency rank and the mixing ratio calculated by each method for the four variants of datasets. In the “w/o special tokens” setting, it was calculated without [CLS] and [SEP].

Methods	Spearman's ρ	
	all tokens (SD)	w/o special tokens (SD)
ATTN-W	0.35 (0.01)	0.35 (0.07)
ATTN-N	-0.23 (0.01)	-0.25 (0.09)
ATTNRES-W	0.35 (0.01)	0.35 (0.07)
ATTNRES-N	-0.79 (0.02)	-0.80 (0.02)
ATTNRESLN-N	-0.36 (0.10)	-0.38 (0.11)

Table 11: Spearman’s ρ between the frequency rank and the mixing ratio calculated by each method for for 25 BERT-base models trained with different random seeds. In the “w/o special tokens” setting, it was calculated without [CLS] and [SEP]. Both of the values are the mean of the values from 25 models, and the standard deviation (SD) is also listed.