Identifying components of BERT that are essential for retrieving linguistic information

Aleksandra Bakalova
Saarland University
alba00013@stud.uni-saarlnd.de

Abstract

In this work we try a new method of localizing components of BERT which are essential for capturing linguistic information. We report the results for BERT-Base model and identify strengths and weaknesses of the approach.
All the code is available on github [1]

1 Idea description and related work

There exist various attempts to understand the way language models work internally. In terms of BERT, the model we consider in this work, there even exists a name for a research subfield focused on its inner workings, which spans hundreds of papers [5]. Previous works have investigated the linguistic knowledge of BERT through different lenses and different methodology, coming sometimes to different results. For example, in [7] authors use a modification of probing technique to claim that basic syntactic information appears in the earlier layers of BERT, while high-level semantic information appears at higher layers. In [4] authors use another probing technique to prove that the localization of linguistic knowledge in BERT is not that simple, and varies across training durations, random seeds and sentences. Such contradictory views are possible because of the methodological limitations: none of the BERTology papers can provide an accurate and complete overview of the BERT inner workings, and can only give some evidence towards one or another hypothesis, but not prove it. All in all, despite the amount of papers written on the topic, there is still room for improvement, and with this work we aim to contribute to the research on localization of BERT's linguistic knowledge.

The goal of this work is to use a new methodology based on importance scores calculation to find the components of BERT model that are responsible for processing different linguistic properties. The idea of importance scores is not new, and has been widely used to find "unimportant components" to prune parameters in convolutional networks [3] and decoder-only transformers [2]. The overall idea of the method is to "knock-out" one of the network's component, i.e. MLP or attention head, and calculate the difference in loss value obtained using original and corrupted network. The higher the difference, the more important is the "knocked-out" component.

One limitation of this approach is the fact that the components in BERT model are highly dependent on each other, thus "knocking-out" one component may highly affect the performance of other. It may result in lower layers being more important than higher layers by design. However, we still think that this approach may shed some light into the BERT inner workings and is worth exploring.

2 Choice of model and tasks

We take BERT-Base-Uncased model from huggingface following many previous works.

[1] https://github.com/CUPalex/bert_importance_scores

Computational Linguistics course project.

For linguistic tasks we take several classification tasks from [1], namely *Sentence Length* (surface-level task), *Tree Depth* (syntactic task), *Top Constituents* (syntactic task), *Tense* (semantic task), *Subject Number* (semantic task), *Object Number* (semantic task). *Sentence Length* is a classification task on the length of the sentence, we use all the 161 different sentence lengths in our dataset as distinct classes. *Tree Depth* is a classification task on the depth of the syntactic tree, which has 16 different classes. *Top Constituents* is a 20-way classification task on a sequence of the constituents in a sentence directly following the Clause constituent. We treat 19 most popular constituent sequences as distinct classes and all the other as single "other" class. *Tense* is a binary classification task on the tense of the verb in main clause, past or present. *Subject Number* is a binary classification task on the number of a subject in the main clause, singular or plural. *Object Number* is a binary classification task on the number of an object in the main clause, singular or plural.

3 Methodology

In this section we describe in detail how our proposed method works.

As already mentioned, the importance score of a component is defined as a difference in loss of the original BERT model and the same model with this component "knocked-out". Component here is either a particular MLP layer, or a particular Self-Attention layer. Thus, BERT-Base model, which consists of 12 layers, has 24 components: 12 MLP layers and 12 Self-Attention layers. "Knocking-out" is performed by substituting the layer weights with Identity matrix, i.e. the matrix which does not change the vector with which it is being multiplied. This ensures that the layer weights in a corrupted network do not impact loss, because they are not used in its computation.

In order to be able to calculate loss for each of the linguistic properties, we first fine-tune BERT model separately for each classification task. In particular, for each of the considered linguistic properties we first freeze the weights of a pre-trained BERT model and then train a linear head on top of it to get predictions for the property. Each of the properties is a sentence-level classification task, and we utilize standard pipeline with training linear head on top of <CLS> token only. The loss used in importance scores calculation is the cross-entropy loss for the considered linguistic task on test dataset.

We train the linear head three times with different random seeds used for random initialization. We train the linear head until the train loss is greater than the average train loss on 5 previous epochs, but for the minimum of 10 epochs and maximum of 50 epochs.

We train the linear head of BERT model on the train split of English Dependency Treebank Universal Dependencies English Web Treebank dataset [6], use validation split for early stopping, and test set for importance scores calculation.

Overall, importance score of component c for property T is defined as follows:

$$\mathcal{IS}_c = |\mathcal{L}(\mathcal{D}, c) - \mathcal{L}(\mathcal{D}, c = \text{Identity})|$$

where $\mathcal{L}$ is the loss defined for task T, $\mathcal{D}$ is the dataset on which we calculate the score,$\mathcal{L}(\mathcal{D}, c$ is the loss of the original BERT model on the given dataset, $\mathcal{L}(\mathcal{D}, c = \text{Identity})$ is the loss of the corrupted BERT model, where the component c is erased, i.e. replaced with the Identity operator which returns the same tensor it receives without any modifications.

In order to validate our approach, we add a random binary task among other linguistic tasks, which is predicting 0 or 1 randomly assigned to each sentence.

4 Results

4.1 Model fine-tuning

In table 4.1 we report the performance of fine-tuned models for for all linguistic tasks and random seeds. We see that for all the tasks, except *Object Number*, the performance of fine-tuned model is significantly higher that chance.

	Random seed	Chance performance	Test accuracy	Epochs of training
Object Number	3	0.77	0.78	14
	2	0.77	0.79	23
	1	0.77	0.78	17
Subject Number	3	0.80	0.90	13
	2	0.80	0.90	15
	1	0.80	0.90	12
Tense	3	0.63	0.87	17
	2	0.63	0.86	17
	1	0.63	0.86	15
Tree Depth	3	0.22	0.52	34
	2	0.22	0.52	30
	1	0.22	0.52	31
Sentence Length	3	0.07	0.26	50
	2	0.07	0.26	50
	1	0.07	0.27	50
Top Constituents	3	0.61	0.78	30
	2	0.61	0.78	33
	1	0.61	0.78	36
Random Task	3	0.50	0.50	12
	2	0.50	0.50	13
	1	0.50	0.50	13

Table 1: Fine-tuning results. For each linguistic property and random seed we report the chance accuracy on test set (portion of the largest class), test accuracy, and number of training epochs.

4.2 Importance scores similarities

First, we look into the similarities between importance scores sequences for different tasks and seeds to ensure that the importance scores sequences for the same task do not change much with different random seeds, ans also to find if some of the tasks are treated similarly in terms of component importance.

In table 4.2 we report average Pearson correlations between importance scores sequences obtained for the same task but with different random seeds or for different tasks. We can see that for all tasks correlations between importance scores obtained from models fine-tuned on the same task are higher on average than between those obtained from models fine-tuned on different tasks. The difference is the biggest for *Random task*, thus we can say that all the considered linguistic tasks are quite similar to each other in terms of component importance. To get a more detailed view on importance scores similarities we plot average Pearson correlations between importance scores sequences obtained from models fine-tuned on each task, without averaging across all other tasks 4.2. We can see that *Tense* task is very different from all others in terms of importance scores correlations, and *Top Constituents, Tree Depth* and *Sentence Length* are very similar to each other. This lies in the paradigm of BERT processing surface, syntactic and semantic differently, since *Top Constituents* and *Tree Depth* are both syntactic tasks, and they are closer to each other than to *Sentence Length*, which is a surface-level task.

4.3 Importance of each layer

We sort the components in each of the fine-tuned models in the order of importance scores and plot the layers to which those components belong in the picture 4.3. We can see that the importance scores for models fine-tuned for the same task but with different random seeds are almost the same for *Sentence Length* and *Top Constituents* tasks. For other tasks they also resemble similar curves.

To get a more detailed view on the localization of important layers for each task, we report the average layer for top-k important components in table 4.3. We can see that for *Top Constituents* and *Tree Depth* the most important components are located at very shallow layers (0-2). For all other tasks the most important layers are also very shallow, *Tense* task has the important components located the deepest among all the tasks, in 4-6 layers. This illustrates the limitation of the method: the shallow

	Within task	Between tasks
Random	0.87	0.04
Object Number	0.91	0.59
Top Constituents	1.00	0.64
Subject Number	0.96	0.58
Tree Depth	0.99	0.67
Tense	0.98	0.38
Sentence Length	1.00	0.70

Table 2: Pearson correlation between importance scores sequences averaged across models fine-tuned for same task but with different random seeds (left column), or across models fine-tuned for different tasks (right column). In more detail, score($task_A$, within) = mean(corr($task_A$-$seed_1$, $task_A$-$seed_2$), corr($task_A$-$seed_1$, $task_A$-$seed_3$), ..., corr($task_A$-$seed_2$, $task_A$-$seed_3$)), score($task_A$, between) = mean(corr($task_A$-$seed_1$, $task_B$-$seed_1$), corr($task_A$-$seed_1$, $task_B$-$seed_2$), ..., corr($task_A$-$seed_1$, $task_C$-$seed_1$), ..., corr($task_A$-$seed_3$, $task_Z$-$seed_3$)). In a sequence of importance scores ith element is an importance score for the ith component, components have some arbitrary but fixed order.

layers might gain more importance because they affect more computational paths. However, this also may serve as some evidence towards the fact that syntactic knowledge is localized in more shallow layers that semantic knowledge.

Another thing that could affect this analysis is the fact that all of the tasks are classification tasks, and there might be very important components that are important for performing classification and not identifying linguistic knowledge. To account for this, we also present for each task and each layer the number of times this layer appeared to be in the top-n most important layers for this task, summed over random seeds and normalized across tasks 4.3. Thus, the higher the score, the more important is the layer to the particular task compared to other tasks.

4.4 Importance of each component

Lastly, we present the analysis of importance of each component for each task by reporting the importance score rank of the component averaged across random seeds 4.4. We can see that Self-Attention in layers 0, 1 and MLP in layers 4, 8 are very important for all linguistic tasks, but not random classification task, Self-Attention in layers 2, 4, 5, 6 and MLP in layer 9 are important for both linguistic tasks and random task. MLP in layer 10 and Self-Attention in layer 11 are very important specifically for *Subject Number* and *Object Number* tasks. Also, most of the components are either quite important for all the tasks or not very important for all the tasks, which supports our correlational analysis and suggests that BERT's internal mechanisms for retrieving linguistic information are quite similar across tasks.

5 Conclusion

In this work we used a new technique based on importance scores calculation to identify the components of BERT that are essential for retrieving linguistic information. We presented evidence that the method is producing meaningful results by ensuring that the model fine-tuning was successful, the correlations of sequences of importance scores for models fine-tuned for the same task but with different random seeds are high, and the results for random task are significantly different than for linguistic tasks. Moreover, we found some evidence supporting the fact that processing of different syntactic properties is done more similarly to each other than to the processing of semantic and surface-level properties. We also performed per-component analysis, identified the most important components for retrieving each linguistic property and found out that these components highly overlap between different linguistic properties.

However, the method presented here has some limitations. First, components of BERT are not independent of each other and removing one of them affects the performance of others. Second, the importance score is not directly sensitive to the importance of component in retrieving linguistic

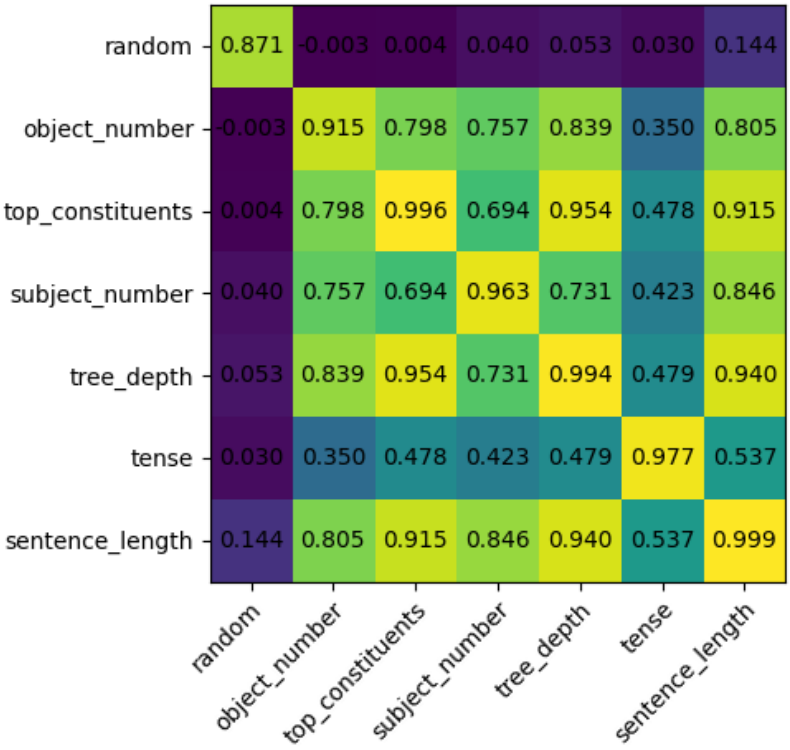

Figure 1: Average Pearson correlation between importance scores sequences obtained from models fine-tuned for each task. The number in the intersection between $task_A$ and $task_B$ is the Pearson correlation between importance scores for $task_A$ and $task_B$ averaged across different combinations of random seeds. In more detail, score$(task_A, task_B)$ = mean(corr$(task_A\text{-}seed_1, task_B\text{-}seed_1)$, corr$(task_A\text{-}seed_1, task_B\text{-}seed_2), \ldots,$ corr$(task_A\text{-}seed_3, task_B\text{-}seed_2)$, corr$(task_A\text{-}seed_3, task_B\text{-}seed_3)$). In a sequence of importance scores ith element is an importance score for the ith component, components have some arbitrary but fixed order.

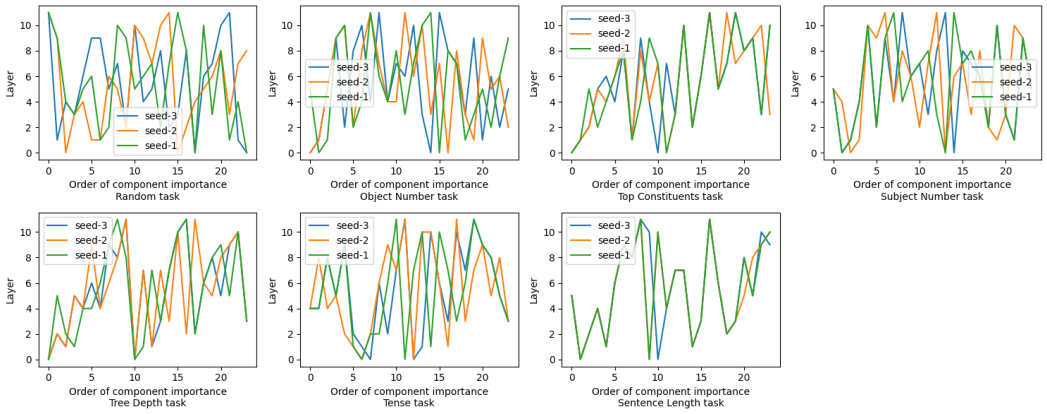

Figure 2: We calculate the importance scores for models fine-tuned for each linguistic task with each random seed and sort the obtained importance scores in descending order. Then we plot the importance orders on the x axis and the layer to which the component with this score belongs on y axis. Thus, the more to the left the component is, the more important it is.

information, but to the importance in performing classification task on linguistically justified classes, thus some of the important components might relate to the importance of class prediction.

		Top-2		Top-3		Top-5		Top-7	
		avg	std	avg	std	avg	std	avg	std
Random	1	10.00	1.00	8.00	2.94	6.40	3.07	5.57	3.20
	2	10.00	1.00	6.67	4.78	5.40	4.03	4.14	3.94
	3	6.00	5.00	5.33	4.19	5.00	3.41	6.14	3.40
	average	8.67	3.54	6.67	4.19	5.60	3.57	5.29	3.63
Object Number	1	2.50	2.50	2.00	2.16	5.00	4.05	4.43	3.58
	2	0.50	0.50	2.00	2.16	5.00	4.05	5.00	3.78
	3	0.50	0.50	2.00	2.16	3.40	3.26	5.00	3.78
	average	1.17	1.77	2.00	2.16	4.47	3.88	4.81	3.72
Top Constituents	1	0.50	0.50	2.00	2.16	2.40	1.85	3.71	2.66
	2	0.50	0.50	1.00	0.82	2.40	1.85	3.86	2.90
	3	0.50	0.50	1.00	0.82	2.80	2.32	3.71	2.66
	average	0.50	0.50	1.33	1.49	2.53	2.03	3.76	2.74
Subject Number	1	2.50	2.50	2.00	2.16	4.00	3.52	4.43	3.58
	2	4.50	0.50	3.00	2.16	4.00	3.52	5.71	4.06
	3	2.50	2.50	2.00	2.16	4.00	3.52	4.43	3.58
	average	3.17	2.27	2.33	2.21	4.00	3.52	4.86	3.80
Tree Depth	1	2.50	2.50	2.33	2.05	2.40	1.85	3.14	2.03
	2	1.00	1.00	1.00	0.82	2.40	1.85	3.57	2.77
	3	1.00	1.00	1.00	0.82	2.40	1.85	3.14	2.03
	average	1.50	1.80	1.44	1.50	2.40	1.85	3.29	2.31
Tense	1	4.00	0.00	5.33	1.89	6.00	2.10	4.43	3.06
	2	6.00	2.00	5.33	1.89	4.60	1.96	3.43	2.50
	3	4.00	0.00	5.33	1.89	6.00	2.10	4.71	2.71
	average	4.67	1.49	5.33	1.89	5.53	2.16	4.19	2.82
Sentence Length	1	2.50	2.50	2.33	2.05	2.40	1.85	3.86	2.90
	2	2.50	2.50	2.33	2.05	2.40	1.85	3.86	2.90
	3	2.50	2.50	2.33	2.05	2.40	1.85	3.86	2.90
	average	2.50	2.50	2.33	2.05	2.40	1.85	3.86	2.90

Table 3: We take top-n most important components according to importance scores ranking, calculate average and standard deviation of the layers to which they belong for each task and random seed. We also report the average and standard deviation across all seeds for each task in the 'average' row.

On the other hand, in contract to probing techniques, this method explores the model's performance in action and not just the information stored in model activations, which might make the findings more easily transferable to other applications.

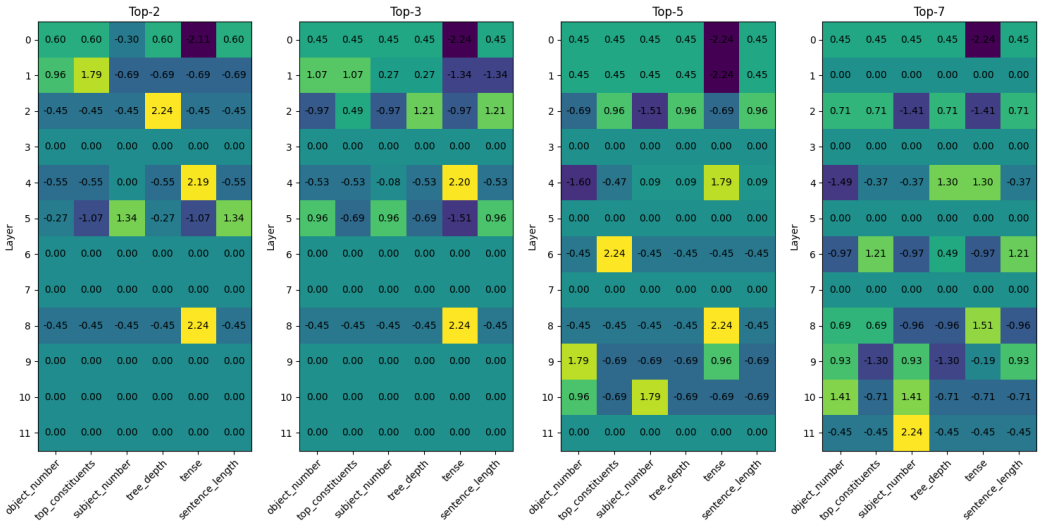

Figure 3: In the intersection of task and layer there is the number of times this layer appeared to be in the top-n most important layers for this task, summed over random seeds and normalized across tasks (substructed mean and divided by standard deviation).

	Random	ObjNum	TopConst	SubjNum	TreeDepth	Tense	SentLen
attn-layer-0	14.00	**0.33**	**0.00**	**1.33**	**0.00**	***6.33***	**1.00**
mlp-layer-0	18.33	15.00	10.67	13.33	10.00	11.67	***9.33***
attn-layer-1	16.00	**1.33**	**1.00**	**2.33**	**2.33**	**5.33**	**4.00**
mlp-layer-1	**4.33**	19.00	***7.00***	20.33	11.67	14.33	14.00
attn-layer-2	***10.00***	***4.67***	**2.33**	***6.67***	**1.33**	***5.67***	**2.00**
mlp-layer-2	12.67	22.00	14.00	18.00	16.67	***7.67***	18.00
attn-layer-3	18.33	12.67	12.00	20.00	13.33	17.00	15.00
mlp-layer-3	**3.00**	18.33	22.33	13.00	23.00	23.00	19.00
attn-layer-4	**2.67**	***8.00***	**4.33**	**2.33**	**4.00**	**1.33**	**3.00**
mlp-layer-4	16.67	***8.67***	***8.67***	***7.33***	***5.67***	**0.00**	11.00
attn-layer-5	***6.33***	**1.33**	**2.67**	**0.00**	**2.33**	**3.00**	**0.00**
mlp-layer-5	13.33	21.33	17.00	23.00	20.00	21.67	20.67
attn-layer-6	***5.33***	10.33	***4.67***	***9.00***	***6.00***	***8.33***	***5.00***
mlp-layer-6	16.00	21.67	15.00	16.00	18.00	16.00	17.00
attn-layer-7	18.33	16.33	18.33	15.33	11.33	10.67	12.00
mlp-layer-7	***10.67***	10.00	10.33	10.33	13.67	17.67	13.00
attn-layer-8	17.33	16.33	20.00	16.00	19.33	21.33	20.33
mlp-layer-8	18.67	***7.00***	***6.67***	10.33	***8.33***	**1.67**	***7.00***
attn-layer-9	***8.33***	20.67	21.00	22.00	20.67	20.00	22.33
mlp-layer-9	**2.67**	**3.00**	***7.67***	***5.67***	***6.33***	***5.67***	***6.00***
attn-layer-10	13.67	12.67	22.67	19.67	22.00	15.33	22.67
mlp-layer-10	12.67	***4.67***	13.00	**4.00**	15.00	13.33	9.67
attn-layer-11	16.67	***7.33***	18.67	***7.00***	16.33	18.33	16.00
mlp-layer-11	**0.00**	13.33	16.00	13.00	***8.67***	10.67	***8.00***

Table 4: The average rank of the importance score for each component and each task. Average is taken across different random seeds. Rank is the position of the component in a sequence of components sorted by importance score. The lower the rank, the more important component is. For each task 5 most important components are in bold, and 5 second-most important components are in bold and italics.

References

1. Alexis Conneau et al. “What you can cram into a single $ &!#* vector: Probing sentence embeddings for linguistic properties”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 2126–2136.

2. Paul Michel, Omer Levy, and Graham Neubig. “Are sixteen heads really better than one?” In: *Advances in neural information processing systems* 32 (2019).
3. Pavlo Molchanov et al. “Pruning Convolutional Neural Networks for Resource Efficient Inference”. In: *International Conference on Learning Representations*. 2016.
4. Jingcheng Niu, Wenjie Lu, and Gerald Penn. “Does BERT rediscover a classical NLP pipeline?” In: *Proceedings of the 29th International Conference on Computational Linguistics*. 2022, pp. 3143–3153.
5. Anna Rogers, Olga Kovaleva, and Anna Rumshisky. “A primer in BERTology: What we know about how BERT works”. In: *Transactions of the Association for Computational Linguistics* 8 (2021), pp. 842–866.
6. Natalia Silveira et al. “A Gold Standard Dependency Corpus for English”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. 2014.
7. Ian Tenney, Dipanjan Das, and Ellie Pavlick. “BERT Rediscovers the Classical NLP Pipeline”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4593–4601.