# Does BERT Rediscover a Classical NLP Pipeline?

**Jingcheng Niu** **Wenjie Lu** **Gerald Penn**
University of Toronto Vector Institute
{niu,luwenjie,gpenn}@cs.toronto.edu

## Abstract

Does BERT store surface knowledge in its bottom layers, syntactic knowledge in its middle layers, and semantic knowledge in its upper layers? In re-examining Jawahar et al. (2019) and Tenney et al.'s (2019a) probes into the structure of BERT, we have found that the pipeline-like separation that they asserted lacks conclusive empirical support. BERT's structure is, however, linguistically founded, although perhaps in a way that is more nuanced than can be explained by layers alone. We introduce a novel probe, called *GridLoc*, through which we can also take into account token positions, training rounds, and random seeds. Using GridLoc, we are able to detect other, stronger regularities that suggest that pseudo-cognitive appeals to layer depth may not be the preferable mode of explanation for BERT's inner workings.

## 1 Introduction

"Surface information at the bottom, syntactic information in the middle, semantic information at the top" (Jawahar et al., 2019). This conclusion is also drawn by Tenney et al. (2019a). While this portrait of multiple layers of linguistic structure has indeed been projected into pipline architectures by some NLP systems before (e.g., Manning et al., 2014), the projection of this understanding onto the internal structure of BERT is now widely accepted both in the BERTology community (Tenney et al., 2019a,b; Hewitt and Liang, 2019; Zhu et al., 2022) and by researchers working on downstream application tasks with BERT (Xiao et al., 2021).

At the same time, there is nevertheless scepticism about the premises of probing itself. As Hewitt and Liang (2019) pondered: "when a probe achieves high accuracy on a linguistic task using a representation, can we conclude that the representation encodes linguistic structure, or has the probe just learned the task?" Furthermore, Pimentel et al. (2020) "cast doubt on whether probing makes sense as a scientific endeavour," because from an information-theoretic perspective, BERT cannot introduce new linguistic information by processing the input sequence.

In our view, this debate does not need to hinder the endeavour of uncovering structure within BERT. Regardless of stance, all parties agree with the existence and importance of different levels of information within BERT itself or its embeddings. There is, however, room for improvement in the investigative methods of both Jawahar et al. (2019) and Tenney et al. (2019a) (J&T), which seem to have been limited to observational confirmations of what they sought to find.

Our own exploratory analysis has revealed that BERT is indeed linguistically founded, although not in a way that suggests a classical pipeline architecture, other than what factors through our own functional understanding of NLP's terminology and subtasks. In addition to examining BERT layers, as J&T did, we have also examined BERT's structure through the lenses of the choice of random seed, training iterations, and, most importantly, token position. We also present several statistical tests of J&T's own conclusions.

We propose *GridLoc*,[1] a self-attention-based probing method that can probe across all of these aforementioned dimensions. Using this novel probing approach and our statistical testing suite, a much more comprehensive picture of the structure of BERT arises. Specifically:

- BERT's task-specific features appear in different token positions in an idiosyncratic but consistent pattern for each task;
- the attending task-specific features exhibit variance across different sentences, different training durations, and different random seeds;
- probe results for *tree depth,* in particular, show an anomalous distribution of linguistic evidence

[1]The implementation, data, plots and results of GridLoc are available online: https://github.com/frankniujc/gridloc_probe and https://doi.org/10.5683/SP3/PCZHN4.



*Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153
October 12–17, 2022.

when taking both layers and token position into consideration.

## 2 Attributing the CNLP to BERT

That there is or was a classical or traditional NLP pipeline is a rather naïve view to take of the history of natural language processing. While there were discussions of independent stages of token-level, syntactic and semantic processing already in the earliest work on machine translation, the pioneers who first engaged with these stages of analysis were of the considered opinion that careful restraint had to be exercised at every level of analysis so that as much of the inherent ambiguity of linguistic input could be carried forward as possible, in the interests of both efficiency and accuracy (Sparck Jones, 2000). It was not until the late 1970s and early 1980s that an excessive reliance on classical logic by the NLP technologists of the grammatico-logical movement, together with contemporaneous psycholinguistics and cognitive science research pointing to a modularity of linguistic structure in human sentence processing (Garrett, 1975, 1980; Fodor, 1983), led to a pipeline-based view that, almost immediately, was apologized for as a convenient abstraction, "incremental" (Levelt, 1989), "highly flexible" and "even opportunistic" (Marslen-Wilson and Tyler, 1987) (see Jackendoff (2000) for a more detailed discussion). While the smoother numerical allocations across tasks in BERTology work, and the distributional graphs drawn by Tenney et al. (2019a) in particular, may at first seem to be commensurate with or even suggest these more nuanced views of a language processing pipeline, the haze of smoke surrounding the visualization of probing evidence makes it extremely difficult to draw precise conclusions from these figures, as we will argue below. Nevertheless, probe methods in general (Adi et al., 2017; Hupkes and Zuidema, 2018; Conneau et al., 2018; Jawahar et al., 2019; Pimentel et al., 2020) are to be credited as one of the few means that we have of approaching the seemingly impossible task of interpreting the neural feature representations within BERT.

Generally, a probe can be *performance-based* or *attention-based*.

### 2.1 Performance-based Probing

A performance-based probe uses an auxiliary task to test for evidence of a particular type of knowledge, by training a supervised classifier with only

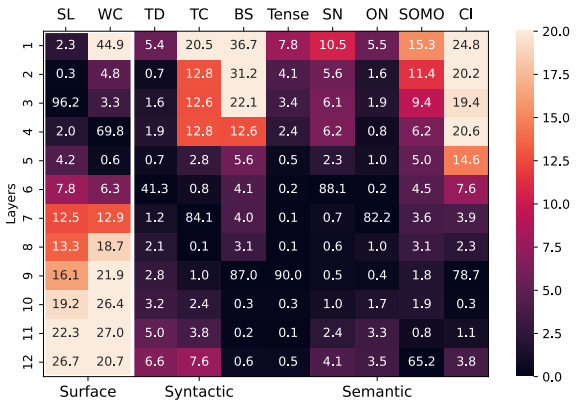



Figure 1: Layer performance probing result of Jawahar et al. (2019), as presented in their Table 2. For clearer visualisation, we transformed this table into a heat map. Each column corresponds to a task, with the best-performing layer in that column containing the performance as a percentage, and the remaining cells displaying their deviation from the best performer in raw percentage points. Surface tasks perform better near the top; syntactic tasks and semantic tasks perform better near the bottom, but their performance patterns are not distinguishable by layer.

BERT's embeddings as input. Good performance of the classifier is interpreted as evidence of relevant linguistic knowledge being present.

Jawahar et al.'s (2019) analysis is a typical application of performance-based probing. They used SentEval (Conneau et al., 2018; Conneau and Kiela, 2018) which contains 10 probing tasks at 3 linguistic levels:

- surface tasks: *sentence length* (SL) and *word content* (WC);
- syntactic tasks: *bigram shift* (BS), *tree depth* (TD) and *top constituent* (TC);
- and semantic tasks: *tense*, *subject number* (SN), *object number* (ON), *semantic odd man out* (SOMO) and *coordination inversion* (CI).[2]

Figure 1 shows the performance of Jawahar et al.'s (2019) probing tasks for each layer. Jawahar et al.'s (2019) visual examination of the results prompted the conclusion that began our introduction. But the observation that semantic information is at the top is puzzling. Except for the SOMO task, all 4 semantic tasks reach peak performance between layers 6 and 9, and all 3 syntactic tasks attain their peak performance within the same range. The similarities between syntactic and semantic

[2]One of our reviewers expressed concerns about why "semantics" and "syntax" were even appropriate labels. Subject/object number and tense, for example, are arguably syntactic/morphological. We agree.

tasks are more apparent with our visualisation of the relative performance differences in Figure 1: the large, dark area in the lower right quadrant is due to an indistinguishability of layers across syntactic and semantic task types.

**Reappraisal of Jawahar et al. (2019)** Some statistical tests may be more illuminating, such as Kendall's (1938) $\tau$ analysis of rank correlation between Jawahar et al.'s (2019) probe performance and a putatively discrete pipeline (surface: 1, syntactic: 2, and semantic: 3). There are two ways to convert Jawahar et al.'s (2019) results into ordinal layer numbers: using a task's top-performing layer number, in which a single number may be selected by multiple tasks (this seems to be what Jawahar et al. (2019) were informally doing), or forcing each task into a distinct layer so that the combined accuracy (the product of all accuracies) of the entire pipeline is maximized. To find this maximal layer assignment, we used SciPy's linear sum assignment to find the maximum sum of logarithms of performance. Both possible layer assignments are shown in Table 1.

The last two columns of Table 1 show the $\tau$ scores. Both layer assignment methods exhibit moderate correlations, but a *post hoc* exclusion of the surface tasks reveals only a weak correlation at the syntactic and semantic levels. These results corroborate our observation that, of the three, only the surface tasks are distinguishable.

### 2.2 Attention-based Probing

Attention has somewhat controversially been interpreted as an explanation of a model's reasoning (Clark et al., 2019). Typically, an attention mechanism assigns a scalar weight to each input source. Hence, the attention mechanism will enhance the important features' effect during training, and so the magnitude of attention weights is often interpreted as an importance score. Clark et al. (2019) and Vig (2019) directly studied the attention mechanism of BERT. Since BERT's attention mechanism does not cross layer boundaries, however, a new probe with an auxiliary attention mechanism is required.

Tenney et al. (2019a) used this attention-based probe to determine which layer contains more task information. Tenney et al. (2019a) exploit a scalar "attention" weight $s_\tau = \text{softmax}(\mathbf{a}_\tau)$ for each layer. The probe classifier is trained using a weighted sum of embeddings (defined as $\mathbf{h}_{i,\tau} = \gamma_\tau \sum_{\ell=0}^{L} s_\tau^{(\ell)} H_i^{(\ell)}$) as input, where $H_i^{(\ell)}$ is the $\ell$-th layer of the $i$-th token's BERT embedding. The value of the attention weights is optimised during the training process, and therefore, the magnitude of the attention weight is understood as a measure of the amount of task-specific knowledge in the corresponding BERT layer.

These scalar weighted embeddings are then leveraged into an improved performance probe that uses *cumulative scoring*. Jawahar et al.'s (2019) probe classifier has access only to a single layer. Tenney et al. (2019a) argue that, since task-specific knowledge can spread out across multiple layers, probing a single layer cannot reveal the full picture. Therefore, they propose to train $L$ probes, each having access to the scalar weighted embeddings from layer 1 to layer $\ell$. The $\ell$-th probe therefore has access to strictly one more layer's worth of information compared to the $(\ell-1)$-th probe. They can then measure the amount of information each layer introduces by calculating the difference between two adjacent probes: $\Delta_\tau^{(\ell)} = \text{Score}_\tau^{(\ell)} - \text{Score}_\tau^{(\ell-1)}$.

They deploy their two tests on the 8 span probing tasks of Tenney et al. (2019b), and aggregate each task's per-layer scalar mixing weights results into a *centre of gravity* score ($\mathbb{E}[\ell] = \sum_{\ell=0}^{L} \ell \cdot s_\tau^{(\ell)}$), and the cumulative scoring results into an *expected layer* score ($\mathbb{E}_\Delta[\ell] = \sum_{\ell=1}^{L} \ell \cdot \Delta_\tau^{(\ell)} / \sum_{\ell=1}^{L} \Delta_\tau^{(\ell)}$).

**Reappraisal of Tenney et al. (2019a)** Again, by visually observing these scores, they conclude that "the tasks [are] encoded in a natural progression: POS tags processed earliest, followed by constituents, dependencies, semantic roles, and coreference." Again, we prefer quantitative tests: while the Pearson's correlation between the centre of gravity and the pipeline ordering of the tasks is weak ($r = 0.319, p = 0.44$), the correlation between expected layer and the pipeline ordering is very strong ($r = 0.933, p = 0.0005$). On the other hand, the correlation between the Kullback-Leibler divergence of the difference scores from a uniform distribution and the pipeline ordering is also very strong ($r = -0.869, p = 0.005$).

### 2.3 Discussion

There are numerous scales along which tasks can be ordered: deep vs. shallow, semantic vs. surface, difficult vs. easy, and over- vs. underdispersed, to name a few. The thesis of Tenney et al. (2019a) is that the first two proceed in lockstep. The last two are highly correlated ($r = -0.879, p = 0.004$).

| Task | SL | WC | TD | TC | BS | Tense | SN | ON | SOMO | CI | $\tau$ | $\tau$ syn sem |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Top | 3 | 4 | 6 | 7 | 9 | 9 | 6 | 7 | 12 | 9 | 0.596 | 0.269 |
| Distinct | 3 | 4 | 6 | 8 | 11 | 5 | 9 | 7 | 12 | 10 | 0.455 | 0.049 |

Table 1: Optimal task layer assignment and Kendall's $\tau$ between the layer assignment and the pipeline (surface, syntactic and semantic information). The "$\tau$ syn sem" column reports Kendall's $\tau$ over only the syntactic task and semantic tasks.

| POS | Const | Deps | Ent | SRL | Coref | SPR | Relns |
|---|---|---|---|---|---|---|---|
| 0.659 | 0.413 | 0.493 | 0.377 | 0.333 | 0.428 | 0.370 | 0.261 |

Table 2: Kendall $\tau$ values of sequences of difference scores, by task, with depth.

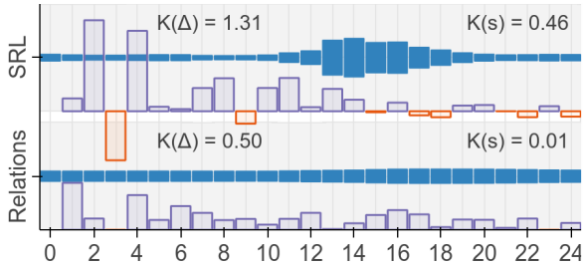



Figure 2: Excerpts of Tenney et al.'s (2019a) layer-wise metrics (Figure 2). Solid (blue) are mixing weights $s_\tau$; outlined (purple) are difference scores $\Delta_\tau$.

The essential problem with Tenney et al.'s (2019a) claims is that their final three, most semantic tasks, are not localizable at all. Because of this, "expected layer," which is really more of a layer torque, is uninterpretable, and not convincingly relatable to layer depth. Even in less dispersed tasks, weighing difference scores by layer number is a statistical fallacy, because layer numbers are ordinal data. Instead, we could look at the positions of the $n$ layers with the highest difference scores, in which $n$ is arbitrary, or analyse the entire sequence of difference scores as a sequence.

Because those scores are not provided in their paper other than in relative terms through unlabelled histograms, what the present authors can do is arrange the ranks of difference scores (1 is highest) by layer (lowest is first) and compute a Kendall's coefficient with respect to the ordered sequence $1, \ldots, 24$ (BERT-large has 24 layers). "Deep" tasks should receive low scores. These coefficients are shown in Table 2. While the Pearson correlation of the pipeline ordering with these scores is strong ($r = -0.793, p = 0.02$), it is not as strong as with the KL divergence of the underlying difference score distributions from uniformity. The claim that BERT mimicks the NLP pipeline is therefore, at best, inconclusive. The empirical data are equally consistent with the counterclaim that BERT is possessed of stripes of surface, syntactic and semantic information that are distributed in parallel throughout its layers, with the semantic information being more evenly dispersed.

The difference score of every probing task in Tenney et al. (2019a) peaks in the first four layers, incidentally, and, in 6 of the 8 tasks, peaks in the first layer.

**Performance is Not a Practical Indicator of Knowledge** Performance-based probes are ill-suited to investigating the structure of BERT, because performance is inherently unstable. Taking the tense task from Jawahar et al.'s (2019) result (Figure 1) as an example; the largest delta between layer 6 (definitely a middle layer) and layer 11 (definitely a top layer) is only 0.3%.

Because Tenney et al.'s (2019a) difference scores are learned, they are not the actual inter-layer deltas of $F_1$ performance. Here, however, each probe has access to exactly one more layer of BERT's contextual representation, and therefore higher layer probes should have access to no less information than lower layer probes. Thus, if probe performance is a good indicator of linguistic knowledge, no higher layer probes should perform worse than lower layer probes. And yet performance drops are prevalent and substantial (Figure 2). Tenney et al. (2019a) suggest that the added new layer introduces distracting features causing the probe classifier to overfit. This means that performance results reflect a combination of knowledge and the probe classifier's ability to generalise — this may be true, but these two variables are hard to separate. Furthermore, neural architectures are stochastic, and so the effect of randomness in performance must also be considered. This is why statistical analysis of observations is crucial to the integrity of the conclusions.

The debate on how to interpret performance in probing is still on-going. Hewitt and Liang (2019) pondered: "when a probe achieves high accuracy on a linguistic task using a representation, can we conclude that the representation encodes linguistic structure, ... has the probe just learned the task?" There are two alternative interpretations of perfor-

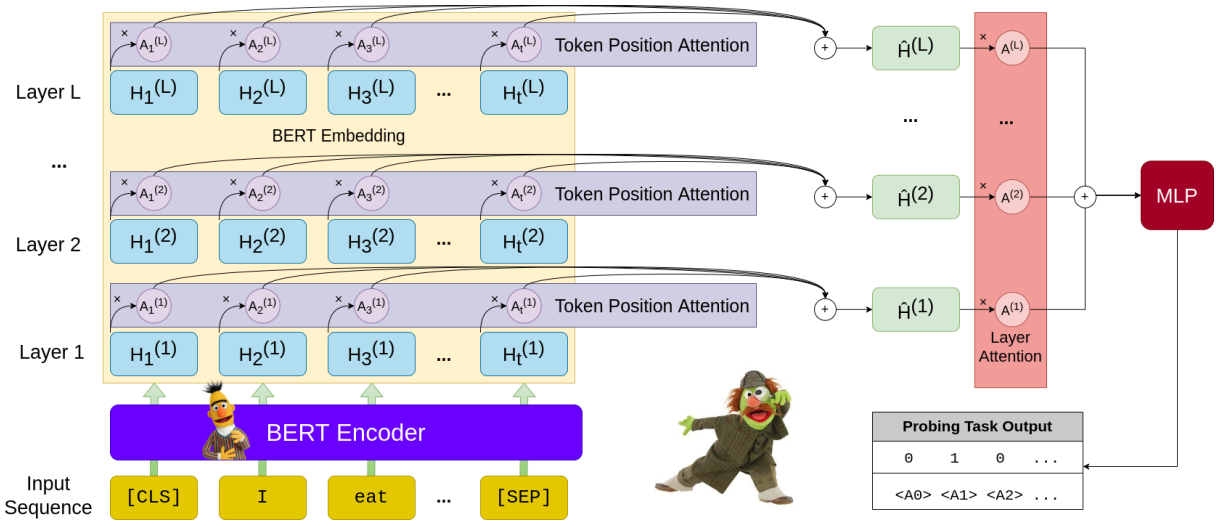



Figure 3: GridLoc model architecture (§3). The BERT-encoded representation of an input sentence first goes through a self-attention pooling process (Lee et al., 2017): an RNN model takes each layer's BERT embeddings $\mathbf{H}^{(\ell)}$ as input and specifically generates an attention weight $\mathbf{A}^{\text{token},(\ell)}$ for that layer. Then the attended representation $\hat{\mathbf{H}}$ goes through another self-attention pooling process, and generates a layer attention weight $\mathbf{A}^{\text{layer}}$. We finally train an MLP classifier that takes the combined representation $\tilde{H}$ as input, and generates a prediction for the probing task. By observing the two attention weights, $\mathbf{A}^{\text{token}}$ and $\mathbf{A}^{\text{layer}}$, we can understand which part of BERT's representation the model assigns importance to.

mance: ease of extraction (Hewitt and Liang, 2019), and mutual information (Pimentel et al., 2020; Pimentel and Cotterell, 2021).

Our reappraisal of Jawahar et al. (2019) and Tenney et al.'s (2019a) results shows that performance is neither intuitively interpretable nor an accurate reflection of knowledge, if performance can be regarded as a reflection at all. It is also a measure entangled with the quality of the probe classifier and randomness.

**Better Control over Attention-based Probing** Attention-based probing is less subject to the aforementioned issues. Firstly, since attention is not the optimisation target, it does not suffer the problem of overfitting. Secondly, although not completely exempt from controversy, attention has generally proven to be a good indicator of feature-importance (Serrano and Smith, 2019; Wiegreffe and Pinter, 2019). Therefore, we can adopt the view of probing results as a reflection of the existence of linguistic, task-specific features. More importantly, since the attention mechanism is purposefully introduced into the probing procedure, we can have a greater degree of freedom and better control over what is probed and where.

Tenney et al. (2019a) used a single shared set of attention weights for every input sentence. This practice cannot capture BERT's variance across sentences (as we will show in §5). But this is not inherent to any limitation of probing techniques in general — the self-attention pooling mechanism (Lee et al., 2017) trains a separate attention network that can assign different attention weights based on the input. Lee et al.'s (2017) self-attention pooling method was originally used by Tenney et al. (2019b) to generate a single span representation over an arbitrarily long span of tokens. The method can also yield a different attention weight for every input sentence. Furthermore, self-attention pooling provides an attention weight distribution over token positions. The similarities between token position attention and layer attention in fact could allow one to analyse the distribution of task knowledge across token positions.

## 3 GridLoc

To leverage all of these degrees of freedom, we present here a novel probing method called *GridLoc*. Figure 3 presents an overview of the probing process. Given an input sentence $S = [t_1, \ldots, t_T]$, BERT produces an $L$-layer embedding for each token $\mathbf{H}_t = [H_t^{(1)}, \ldots, H_t^{(L)}]$. GridLoc can produce a more complete picture of where task specific knowledge resides, by breaking down the probe's attention weight across both token positions and layers, as well as across random seeds and training iterations.

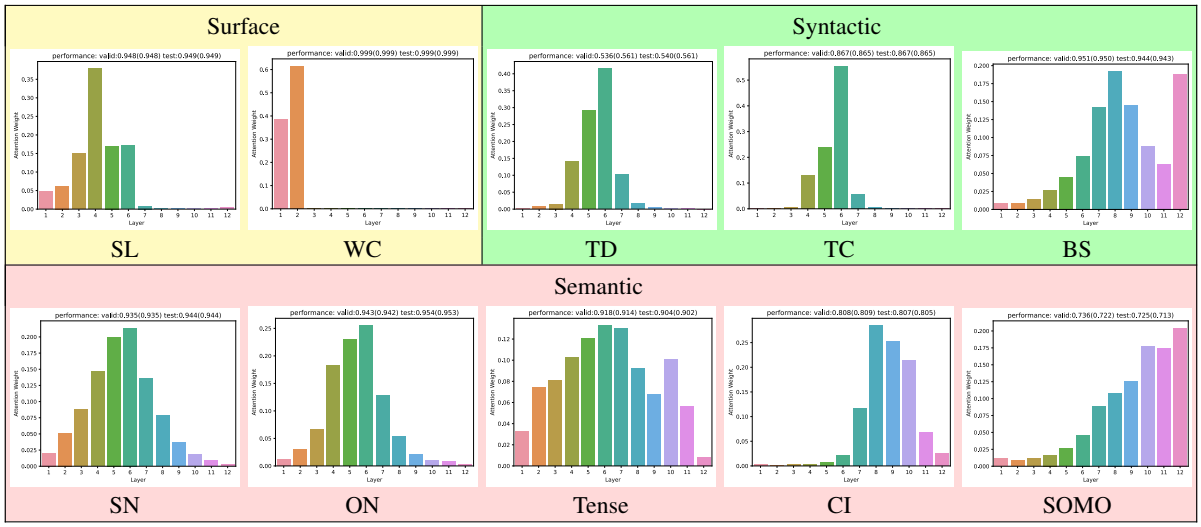
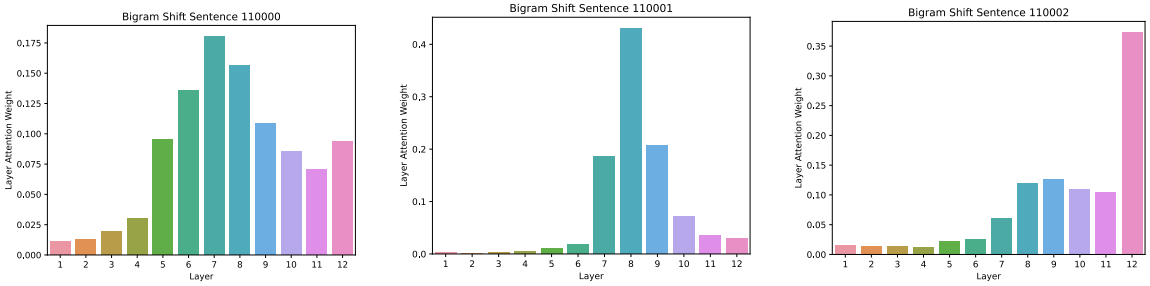
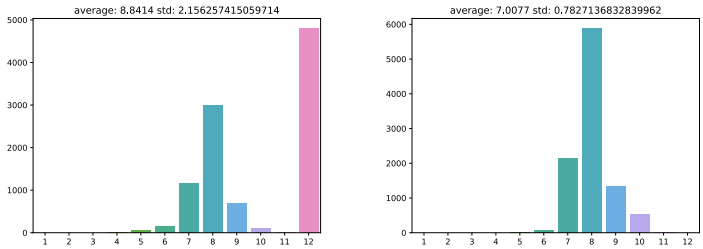



Figure 4: GridLoc average layer attention weight distribution for every SentEval task. For each task, the plot shows the average layer attention weight of all of the test set sentences during the top-performing epoch (by validation set) with random seed 0. We observed a moderate $\tau = 0.503$ with the entire pipeline (surface + syntactic + semantic), but a mere $\tau = 0.134$ with only the syntactic and semantic tasks. GridLoc confirms our earlier observation: surface tasks attend to lower layers, but syntactic and semantic tasks are inseparable.




(a) Layer attention weights of the same probe of the first three Bigram Shift test-split sentences. The layer attention weight distributions differ widely.



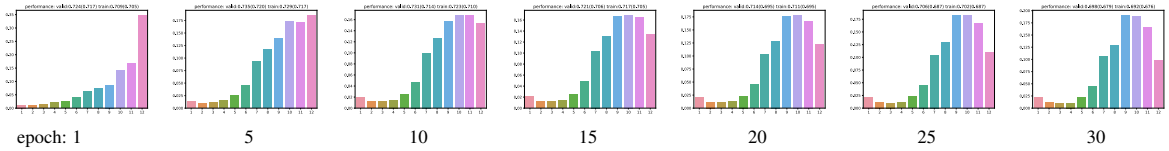

Seed: 0, Epoch: 7. Seed: 1, Epoch: 8.

(b) Distribution of the layer with the highest attention score over the Bigram Shift test-set sentences for two probing runs with different random seeds. Both probes are generated at their top performing (by validation) epochs. Distributions can exhibit substantial variance (left: $\sigma = 2.16$, right: $\sigma = 0.78$). For the run with seed 0 (left), there is also a spike in sentences at the 12th layer that is not observed in the run with seed 1 (right).

| Task | $\sigma$ |
|---|---|
| SL | 1.468 |
| WC | 0.786 |
| BS | 1.95 |
| TD | 0.584 |
| TC | 1.025 |
| Tense | 2.359 |
| SN | 1.188 |
| ON | 0.903 |
| SOMO | 1.589 |
| CI | 0.953 |

(c) Standard deviation of the distribution of the layer with the highest attention score of every SentEval probing task.

Figure 5: Variance of probing results among sentences.



epoch: 1 5 10 15 20 25 30

Figure 6: An example (SOMO with random seed 0) of the average attention weight distribution change over training iterations.

**Token Position** To understand the task-specific feature distribution across token for each layer, we exploit the aforementioned self-attention pooling (Lee et al., 2017) method to learn a *token attention:*

$$\mathbf{A}^{\text{token},(\ell)} = \text{softmax}(\mathbf{w}_{\text{token}} \cdot \text{RNN}(\mathbf{H}^{(\ell)}))$$

specific to each layer of embeddings $\mathbf{H}^{(\ell)} = [H_{t_1}^{(\ell)}, \ldots, H_{t_T}^{(\ell)}]$. Then, we can obtain an attended hidden representation:

$$\hat{\mathbf{H}} = \mathbf{A}^{\text{token}} \cdot \mathbf{H}$$

**Layer** Then, we learn a sentence-specific *layer attention:*

$$\mathbf{A}^{\text{layer}} = \text{softmax}(\mathbf{w}_{\text{layer}} \cdot \hat{\mathbf{H}}^{(\ell)})$$

from the attended contextual embedding of the entire sequence. Finally, we can train the probe classifier on the fully attended representation of the input sequence:

$$\tilde{H} = \mathbf{A}^{\text{layer}} \cdot \hat{\mathbf{H}}$$

by attending to both the tokens and the layers.

**Randomness and Training** To understand the variance of the probe result relative to the random seed, we repeat each of our experiments with 20 seeds ($0 \sim 19$). We also maintain a record of 30 epochs of training for each probe.

## 4 Experimental Setup

We used all 10 tasks in SentEval (Conneau and Kiela, 2018) as described in §2.1. To be consistent with J&T's results, we conducted our experiment using the uncased BERT-base model and Jawahar et al.'s (2019) hyperparameters.[3]

## 5 Experimental Results

### 5.1 Layers Alone do Not Recapitulate the Pipeline

With our new probe, we calculate the average layer attention weight for each task and report in Figure 4 an example for every task. The average layer attention weight is calculated by summing up every test sentence's layer attention weight, and then normalising by the size of the test set. The average task layer attention weight is a good global indicator of the spread of task-specific features in BERT. Our results agree with our observations based upon Jawahar et al. (2019): although surface task features are dense in lower layers, and both syntactic and semantic task features are spread out between mid to upper layers; the inseparable syntactic and semantic tasks show that BERT layers alone do not recapitulate the putative pipeline.

This observation is corroborated by a Kendall's $\tau$ test between every run (20 random seeds $\times$ 30 epochs) of each task's top performing layer and the 1–3 pipeline-based ranking described in §2.1.[4] Since now we have 600 data points for each task, our correlation test result is more robust. Again, we observed a moderate $\tau = 0.503$ with the entire pipeline (surface + syntactic + semantic), but a mere $\tau = 0.134$ with only the syntactic and semantic tasks.

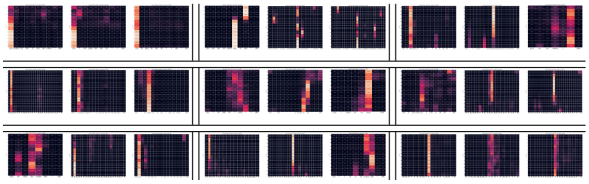


Figure 7: Token-position attention-weight plots for the first 3 sentences of the SentEval test set on all SentEval tasks (from top left to bottom right: SL, WC, Tense, SN, ON, BS, TC, SOMO and CI). The attention weights are displayed as a 2-dimensional heat map; each column corresponds to a token and each row corresponds to a BERT layer. Brighter colours represent larger attention weights. For most sentences, the token-position attention at every layer attends to the same token, hence the bright vertical line.

| POS | Count | Top Layer |
|---|---|---|
| PUNCT | 23402 | 7.18 |
| NOUN | 19077 | **7.33** |
| VERB | 18277 | 5.03 |
| PRON | 16120 | 6.68 |
| ADP | 11129 | **3.19** |

(a) Average best-attending layer of the 5 most common POSs. Maximum and minimum are highlighted in bold.

| POS 1 | POS 2 | $\lvert$PB $r\rvert$ |
|---|---|---|
| PUNCT | ADP | 0.483 |
| ADJ | ADP | 0.462 |
| ADP | DET | 0.460 |
| NOUN | ADP | 0.438 |
| PRON | ADP | 0.399 |

(b) Best 5 absolute point-biserial correlations between the best attending layers of tokens with different POSs. $p$-values are less than $10^{-323}$.

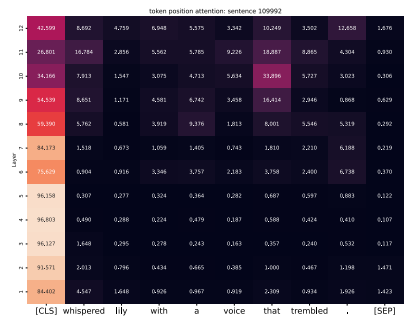
Table 3: The tree depth probe attends to tokens with different POS at different layers.

### 5.2 Variance through Sentences, Randomness and Training Time

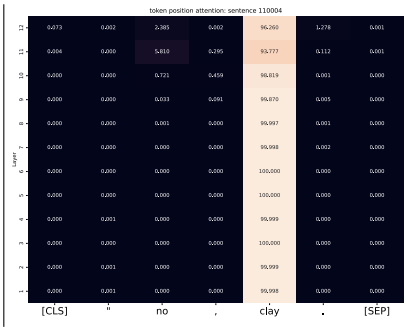
Nevertheless, average attention weight is a global measure that withholds important nuances regarding the variance of the probe along several dimensions. As shown in Figure 5a, layer attention as-

[3] https://github.com/ganeshjawahar/interpret_bert

[4] The unique ranking test is discarded as it cannot generalise to our situation with multiple runs.




(**SL**) The probe classifier attends to the `[CLS]` token, as it is often regarded as the "embedding of the sentence," and length is a global feature of the sentence.

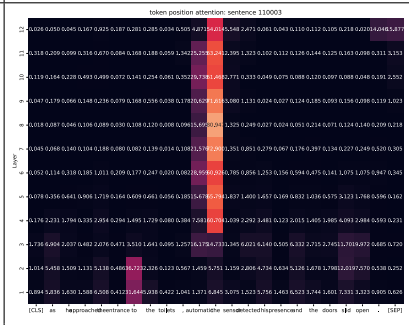



(**WC**) The probe classifier attends to the target word. In this particular example, every token-position weight attends to the target token "clay."


(**Tense**) The probe classifier attends to the verb or its tense-morphology-bearing "wordpiece" (Wu et al., 2016), such as `##ed` or `##es`.


(**SN**) The probe classifier attends to the subject noun or its number-morphology-bearing "wordpiece."

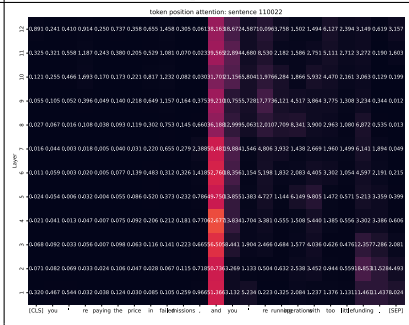

(**ON**) Similar to SN, the classifier attends to the object noun of the sentence or its number-morphology-bearing "wordpiece."




(**BS**) The probe classifier attends to the words that are being inverted. In this case, the two inverted words are "corners" and "the." If an original sentence is encountered, the classifier will place heavier weights on places where inversions are noticeable, such as prepositions, determiners and punctuation.


(**TC**) Top constituents attend to the first one or two words in the sentence, as tags with one presentential modifier followed by NP VP are commonplace. E.g., in this case, the sequence is labelled as `RB_NP_VP_`.


(**SOMO**) The probe classifier attends to the verb or noun (here, the verb "confused") that is replaced. Upon encountering an unaltered sentence, the classifier will attend to common verbs and nouns that are likely to be replaced.




(**CI**) The probe classifier attends to the coordinating conjunction (CC). CC is crucial in determining whether the sentence has inverted coordination since different CCs serve different purposes when connecting two parts of a sentence.

Figure 8: Example token-position attention plots and their pattern analyses. Similar to Figure 7, the attention weight is displayed in a 2-dimensional heat map, with larger weights associated with brighter colours. The tokens of the example sentence are displayed along the x axis. The number on each cell is the attention weight as a percentage. Since attention weight is normalised by softmax at each layer, numbers in every row should sum up to 100.




Figure 9: Tree Depth token-position attention weights. The token-position attention tends to focus on tokens with different parts-of-speech at different layers. For instance, prepositions such as *to*, *of*, *on* and *at* have higher attention weights at lower layers, and punctuation has higher attention weights at higher layers.

signs drastically different weights to different input sentences. This observed variance is not idiosyncratic. Figure 5b aggregates counts of the layer with the highest attention score over the test set sentences. The difference between seed 0 and seed 1 also emphasises that probe results are not immune to random initialization effects. This high variance is also not unique to Bigram Shift. Table 5c shows the overall standard deviation of every SentEval task. In all but two tasks, the standard is one full layer or more. Figure 6 shows how average attention weight can change during training. In this particular run, the distribution does not stabilize until about epoch 15. This again demonstrates the importance of utilising the self-attention pooling mechanism, which enables us to capture these variations.

### 5.3 Consistently Idiosyncratic Token Positions

Contrary to the variance we observed by layer, our token-position attention results are more stable — *tree depth* is the only exception, and we will discuss it shortly. As indicated in Figure 7, almost every sentence's token-position attention focuses on the same token in every layer. The choice of that token position is not arbitrary — there are linguistic reasons for them. Figure 8 shows one example for every task along with our analysis.

### 5.4 Tree Depth: an Insightful Anomaly

The token-position attention-weight result of the *tree-depth* task is the only exception to the bright vertical line pattern. Here, the probe attends to multiple tokens at different layers (Figure 9). The attention patterns are not arbitrary, however. As shown in Table 3a, tokens with different parts-of-speech[5] (POS) receive the most attention from the probe at different layers. Among the tokens, nouns attend to the highest layers at 7.33 (middle) and prepositions attend to the lowest layers at 3.19 (low). To verify the significance of the mapping between POS and layer attention, we conducted point-biserial correlation tests (Table 3b). We observed a moderate correlation between several pairs of POS, confirming that the probe can discriminate between them in this manner.

Although how any of this might relate to tree depth is unclear, this finding is still important in two ways. First, the Damoclean sword that the probe classifier is merely able to generalise the probing corpus (Hewitt and Liang, 2019; Pimentel et al., 2020) has been removed from over the claim that BERT is in possession of linguistic knowledge. POS information is certainly not self-evident in a corpus labelled with tree depth. Here we have a probe on one auxiliary linguistic task attesting to another linguistic phenomenon.

Second, the distribution of linguistic features defies J&T's proposed distribution of knowledge in BERT. What we see here is not different levels of linguistic knowledge from the pipeline occupying different layers of BERT, but rather different information from the same pipeline level (i.e., distinct POS labels) occupying different layers of BERT. This insight would not have been available without a probe that takes both the token position dimension and the layer dimension into account.

## 6 Conclusion

Did BERT rediscover an NLP pipeline? Not in a naïve, architectural sense. GridLoc reveals a structure in BERT that is more intricate than a flowchart of a pipeline could accurately portray, and yet it does seem to be linguistically founded. We find that probing results regarding BERT layers are unstable, diverging across sentence input, random seeds and the early iterations of training. The distribution of linguistically motivated task features along token positions, on the other hand, is relatively more stable. Moreover, GridLoc's results on tree depth provide preliminary evidence of POSs being used to conduct novel but linguistically generalizable inference concerning a derivative syntactic phenomenon.

## Acknowledgements

We thank Zining Zhu (University of Toronto) for all the insightful discussion. We also want to thank the anonymous reviewers for providing informative comments and suggestions.

## References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT

[5]Generated by the Stanza (Qi et al., 2020) package.

Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings LREC 2018*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings ACL 2018*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jerry A. Fodor. 1983. *The Modularity of Mind*. The Modularity of Mind. The MIT Press, Cambridge.

M. F. Garrett. 1975. The Analysis of Sentence Production. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 9, pages 133–177. Academic Press.

M. F. Garrett. 1980. Levels of Processing in Sentence Production. In B. Butterworth, editor, *Language Production*, volume 1. Academic Press, London.

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings EMNLP-IJCNLP 2019*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Dieuwke Hupkes and Willem Zuidema. 2018. Visualisation and 'Diagnostic Classifiers' Reveal how Recurrent and Recursive Neural Networks Process Hierarchical Structure (Extended Abstract). In *Proceedings IJCAI 2018*, pages 5617–5621, Stockholm, Sweden. International Joint Conferences on Artificial Intelligence Organization.

Ray Jackendoff. 2000. Fodorian Modularity and Representational Modularity. In Yosef Grodzinsky, Lewis P. Shapiro, and David Swinney, editors, *Language and the Brain*, pages 3–30. Elsevier.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings ACL 2019*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

M. G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *Proceedings EMNLP 2017*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

W. J. M. Levelt. 1989. *Speaking: From Intention to Articulation*. ACL-MIT Press Series in Natural-Language Processing. MIT Press, Cambridge, Mass.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings ACL 2014: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

William Marslen-Wilson and Lorraine K. Tyler. 1987. Against Modularity. In Jay L. Garfield, editor, *Modularity in Knowledge Representation and Natural-Language Understanding*, pages 37–62. The MIT Press.

Tiago Pimentel and Ryan Cotterell. 2021. A Bayesian Framework for Information-Theoretic Probing. In *Proceedings EMNLP 2021*, pages 2869–2887, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-Theoretic Probing for Linguistic Structure. In *Proceedings ACL 2020*, pages 4609–4622, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings ACL 2020: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? In *Proceedings ACL 2019*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Karen Sparck Jones. 2000. R.H. Richens: Translation in the NUDE. In W.J. Hutchins, editor, *Early Years in Machine Translation*, pages 263–278. John Benjamins.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings ACL 2019*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Jesse Vig. 2019. A Multiscale Visualization of Attention in the Transformer Model. In *Proceedings ACL 2019: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. In *Proceedings EMNLP-IJCNLP 2019*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

Zeguan Xiao, Jiarun Wu, Qingliang Chen, and Congjian Deng. 2021. BERT4GCN: Using BERT Intermediate Layers to Augment GCN for Aspect-based Sentiment Classification. In *Proceedings EMNLP 2021*, pages 9193–9200, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zining Zhu, Soroosh Shahtalebi, and Frank Rudzicz. 2022. Predicting fine-tuning performance with probing. In *Challenges & Perspectives in Creating Large Language Models*.