Are All Neurons Created Equal? Interpreting and Controlling BERT through Individual Neurons

Omer Antverg
Technion – Israel Institute of Technology
omer.antverg@cs.technion.ac.il

Yonatan Belinkov
Technion – Israel Institute of Technology
belinkov@technion.ac.il

Abstract

While many studies have shown that linguistic information is encoded in hidden word representations, few have studied individual neurons, to show how and in which neurons it is encoded. Among these, the common approach is to use an external probe to rank neurons according to their relevance to some linguistic attribute, and to evaluate the obtained ranking using the same probe that produced it. We show that this methodology confounds distinct factors—probe quality and ranking quality—and thus we separate them. We compare two recent ranking methods and a novel one we introduce, both by probing and by causal interventions, where we modify the representations and observe the effect on the model's output. We show that encoded information and used information are not always the same, and that individual neurons can be used to control the model's output, to some extent. Our method can be used to identify how certain information is encoded, and how to manipulate it for debugging purposes.[1]

1 Introduction

What linguistic information is encoded in a hidden representation? A popular approach to tackle this question is by probing: training a simple model to predict some information from the representation [Adi et al., 2017, Conneau et al., 2018]. Most studies focus on training the classifier using the entire high-dimensional representation to predict the desired attribute. Such a method can show whether the information is encoded in the representation, but not **how** it is encoded: in which neurons? is it localized (concentrated in a small set of neurons) or dispersed? A few recent studies [Elazar et al., 2021, Feder et al., 2020] have tackled a related, causal question: given that the information is encoded, is it actually being used by the language model that encoded it? If we want to alter a model's behaviour—for example, for debugging purposes—we would like to know both how the information is encoded and whether it is used by the model. Such knowledge may allow us to modify specific neurons of the representation in a certain way that would change the model's output, with respect to that attribute—as in Bau et al. [2019]—and provide us with parameters-level explanations of the model's decisions. Other reasons to look at individual neurons can be to reduce the number of components in the model, using methods such as pruning [Voita et al., 2019, Sajjad et al., 2020], as well as gain a general scientific understanding of the model.

In this work, we focus on two prior neuron ranking methods, which aim to rank the neurons of a word representation according to their importance for a linguistic property, such as part of speech. Both of these methods—LINEAR [Dalvi et al., 2019] and GAUSSIAN [Torroba Hennigen et al., 2020]—rely on an external probe to obtain a ranking: the first makes use of the internal weights of a linear probe, while the second considers the performance of a decomposable generative probe.

[1] Our code is available at: https://github.com/technion-cs-nlp/Individual-Neurons

1st Workshop on eXplainable AI approaches for debugging and diagnosis (XAI4Debugging@NeurIPS2021).

Both methods evaluate the quality of their ranking according to the probe's accuracy using only the k-highest ranked neurons. However, using this metric, two distinct factors are conflated: the probe's quality as a classifier, and the quality of the ranking it produces. We claim that we should separate the two: a good classifier may provide good results even if its ranking is bad, and an optimal ranking may cause an average classifier to provide better results than a good classifier that is given a bad ranking. To avoid this conflation, we propose a simple ranking method, PROBELESS, which ranks neurons according to the difference in their values across labels, and thus can be derived directly from the data, with no probing involved.

To disentangle probe quality and ranking quality, we use a probe from one method with a ranking from another one. We find that while GAUSSIAN generally provides higher accuracy, its selectivity [Hewitt and Liang, 2019] is lower, implying that it relies more on memorization, which improves probing quality but not necessarily ranking quality. We further find that GAUSSIAN provides the best ranking for small sets of neurons, while LINEAR provides a better ranking for large sets.

To apply a comparison that focuses on the rankings and is free of probing, and to find which ranking selects neurons that are used by the model, we apply interventions on the representation: we modify subsets of neurons from each ranking and observe the effect on language modeling w.r.t to the property in question. We find that PROBELESS tends to select neurons that are used by the model, more so than the two probing-based rankings.

We introduce a novel intervention method that allows a fine-grained control of the model's output with respect to a linguistic attribute, to some extent. For example, intervening on the tense attribute in English may turn the words "using" and "think" into "used" and "thought". This method can be used as a neuron-level debug tool, allowing developers to understand how easy or hard it is to reduce or magnify some of the model's properties, e.g., gender bias, by observing how the gender attribute is spread along the neurons and the success rate of changing a word's gender. By comparing probing results and intervention results, we deduce that there is an overlap between encoded information and used information, but they are not the same.

We experimentally analyze the M-BERT model [Devlin et al., 2019] on 9 languages and 12 morphological attributes, from the Universal Dependencies dataset [Zeman et al., 2020]. Our experiments reveal the following insights:

- We show the need to separate between probing quality and ranking quality, by showing cases where intentionally poor rankings provide better accuracy than good rankings, due to probing weaknesses.
- We present a new ranking method that is free of any probes, and tends to prefer neurons that are being used by the model, more so than existing probing-based rankings.
- We show that there is an overlap between encoded information and used information, but they are not the same.
- We show the advantage of looking into individual neurons, especially for debugging purposes, to allow a fine-grained control over the model's output.

2 Neuron Rankings and Data

We begin by introducing some notation. We denote the word representation space as $H \subseteq \mathbb{R}^d$ and an auxiliary task as a function $F : H \to Z$, for some task labels Z (e.g., part-of-speech labels). Given a word representation $h \in H$ and some subset of neurons $S \subseteq \{1, ..., d\}$, we use h_S to denote the subvector of h in dimensions S. For some auxiliary task F and $k \in \mathbb{N}$, we search for an optimal subset S^* such that $|S^*| = k$ and h_{S^*} contains more information regarding F than any other subvector $h_{S'}, |S'| = k$. For the search task, we define *neuron-ranking* as a permutation $\Pi(d)$ on $\{1, ..., d\}$ and consider the subset $\Pi(d)_{[k]} = \{\Pi(d)_1, ..., \Pi(d)_k\}$. Our goal is to find an optimal ranking $\Pi^*(d)$ such that $\forall k, \Pi^*(d)_{[k]}$ is the optimal subset with respect to F. However, finding such an optimal ranking, or even an optimal subset, is NP-hard [Binshtok et al., 2007]. Thus, we focus on several methods to produce rankings, which provide approximations to the problem, and compare them.

2.1 Rankings

The ranking methods we compare include two rankings obtained from previously suggested probing-based neuron-ranking methods, and a novel ranking we propose, based on data statistics rather than probing.

2.1.1 LINEAR

The first method, henceforth LINEAR (named linguistic correlation analysis in Dalvi et al. [2019]), trains a linear classifier on the representations to learn the task F. Then, it uses the trained classifier's weights to rank the neurons according to their importance for F. They showed that their method identifies important neurons through probing and ablation studies, and found that while the information is distributed across neurons, the distribution is not uniform, meaning it is skewed towards the top-ranked neurons. In Dalvi et al. [2019], after the probe has been trained, its weights are fed into a neuron ranking algorithm. However, we observed that the proposed algorithm distributes the neurons equally among labels, meaning that each label would contribute the same number of neurons at each portion of the ranking, regardless of the amount of neurons that are actually important for this label. Thus, we chose a different way to obtain the ranking: for each neuron, we compute the mean absolute value of the $|Z|$ weights associated with it, and sort the neurons by this value, from highest to lowest. In early experiments we found that this method empirically provides better results, and is more adapted to large label sets.

2.1.2 GAUSSIAN

The second method, henceforth GAUSSIAN [Torroba Hennigen et al., 2020], trains a generative classifier on the task F, based on the assumption that each dimension in $\{1, ..., d\}$ is Gaussian-distributed. Then, it makes use of the decomposability of the multivariate Gaussian distribution to greedily select the most informative neuron, according to the classifier's performance, from $\{1, ..., d\}$ at every iteration. This way we greedily obtain a full neuron ranking after training only once, while applying this greedy method to LINEAR would require retraining the probe $d!$ times, which is clearly infeasible. They found that most of the tasks can be solved using a relatively low number of neurons, but also noted that their classifier is limited due to the Gaussian distribution assumption.

2.1.3 PROBELESS

The third neuron-ranking method we experiment with is based purely on the representations, with no probing involved, making it free of probing limitations [Belinkov, 2021] that might affect ranking quality. For every attribute label $z \in Z$, we calculate $q(z)$, the mean vector of all representations of words that possess the attribute and the value z. Then, we calculate the element-wise difference between the mean vectors,

$$r = \sum_{z,z' \in Z} |q(z) - q(z')|, \qquad r \in \mathbb{R}^d \tag{1}$$

and obtain a ranking by arg-sorting r, i.e., the first neuron in the ranking corresponds to the highest value in r. In the case of a binary-valued attribute, this is simply the difference in means. For attributes with more than two values, PROBELESS prefers neurons that separate different labels over neurons that have a similar value across labels.

2.2 Data

Throughout our work, we follow the experimental setting of Torroba Hennigen et al. [2020]: we map the UD treebanks [Zeman et al., 2020] to the UniMorph schema [Kirov et al., 2018] using the mapping by McCarthy et al. [2018]. We select a subset of the languages used by Torroba Hennigen et al. [2020]: Arabic, Bulgarian, English, Hindi, Finnish, French, Russian, Spanish and Turkish, to keep linguistic diversity. We work with morphological attributes from these languages. Full data details are provided in Torroba Hennigen et al. [2020] and further data preparation steps are detailed in Appendix A.1. We process each sentence in a pre-trained M-BERT, and take word representations from layers 2, 7 and 12 of the model, to see if there are different patterns in the beginning, middle and end of the model. We end up with a total of 156 different configs (language, attribute, layer) to test. For words that are split during tokenization, we define their final representation to be the average over their sub-token representations. Thus, each word has one representation for each layer, of 768 dimensions. We do not mask any words throughout our work.

3 Probing

Given some ranking $\Pi(d)$, we would like to evaluate how well it sorts the neurons for the task F. Our first ranking-evaluation approach is the standard probing approach from previous work [Dalvi et al., 2019, Torroba Hennigen et al., 2020], where we expose the classifier to a subvector of the representation and evaluate how well it predicts the task. However, while previous work conflated rankings and classifiers, we are more careful: we separate the two, and pair each ranking with two classifiers, meaning that at least one of them is completely unrelated to the ranking.

Formally, for an increasing $k \in \mathbb{N}$, we train a classifier $f_\theta : H_k \to Z$ with parameters θ to predict the task label, $F(h)$, solely from $h_{\Pi(d)_{[k]}}$ (the subvector of the representation h in the top k neurons in ranking Π), ignoring the rest of the dimensions.[2] The general assumption is that the better f_θ performs, the more task-relevant information is encoded in $h_{\Pi(d)_{[k]}}$. Yet, the behaviour of f_θ itself might affect results and conclusions about the ranking. Hence, we aim for a fine-grained analysis of our results, and also measure selectivity (§ 3.1.1).

3.1 Experimental Setup

As classifiers, we experiment with both classifiers used by the first two ranking methods (§ 2.1.1, 2.1.2). As rankings, we experiment with the 3 ranking methods described in 2.1. For each, we use the original ranking it produces and its reversed version, and refer to them as top-to-bottom and bottom-to-top, respectively. To those we add a random ranking baseline, ending up with 7 different rankings overall. We compare all classifier-ranking combinations for the same k.

Since both of the first two ranking methods are inherently tied to the classifier that was used to generate them, and the third ranking is a classifier-neutral ranking, it can be used for a fair comparison between the classifiers.

3.1.1 Metrics

Accuracy First, we measure the accuracy of the probe's predictions. Since we experiment with many different configs, we use the Wilcoxon signed-rank test [Wilcoxon, 1992] as a statistical significance test to determine whether a certain combination of a classifier and a ranking is statistically significantly better than another combination.

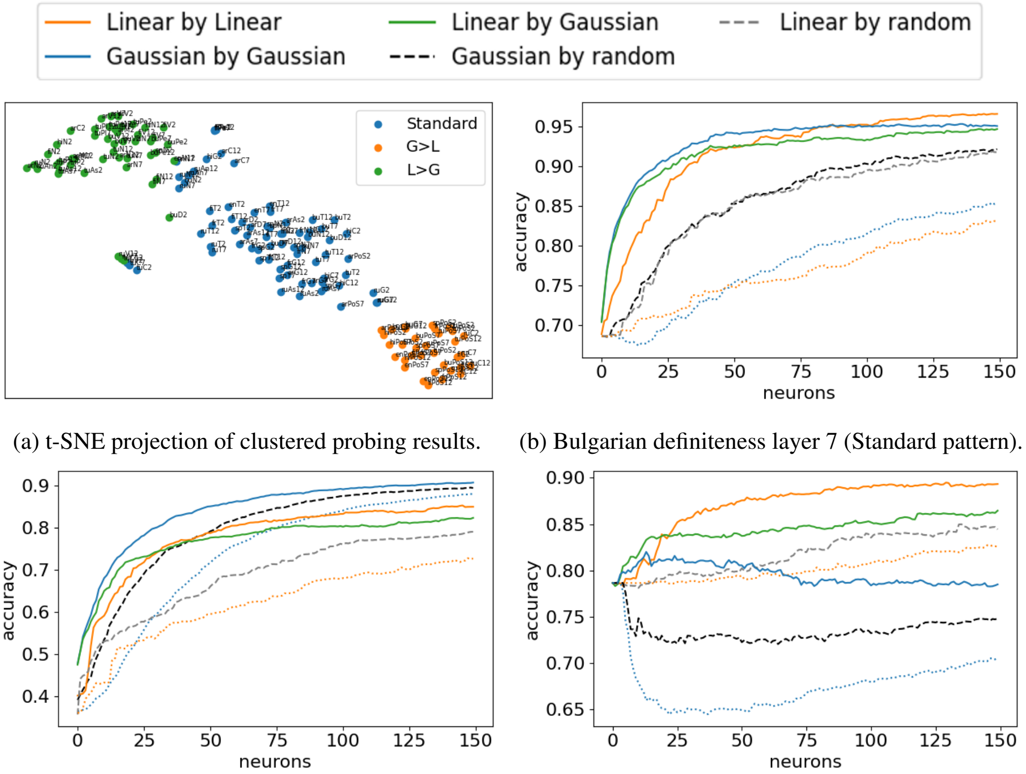
Selectivity We also evaluate our probes by selectivity [Hewitt and Liang, 2019], which is defined as the difference between the classifier's accuracy on the actual probing task and its accuracy on predicting random labels assigned to word types, called a control task. Low selectivity implies that the probe is capable of memorizing the word-type–label pair, and so high accuracy in the probing task does not necessarily entail the presence of the linguistic attribute. Thus, we prefer probes that are both accurate and selective.

3.2 Results

Fig. 1a shows a t-SNE [van der Maaten and Hinton, 2008] projection after performing K-means clustering on our 156 accuracy results from the configs we tested, where each point represents accuracy results from one config (details on clustering procedure are in Appendix A.2). It shows three clusters of configs, which correspond to three distinct patterns, demonstrated in Figs. 1b-1d. In these figures, each color represents a combination of a classifier and a ranking, where a solid line is used for the top-to-bottom version of the ranking and a dotted line is for the bottom-to-top version of it, and a dashed line is used for random ranking.

Almost half of the configs follow the Standard pattern (Fig. 1b), in which all top-to-bottom rankings are always better than the random ranking, which is always better than all bottom-to-top rankings. In

[2] We only probe into representations of words that possess the attribute, e.g., if the attribute is gender we do not probe into the representation of the word "pizza".



(a) t-SNE projection of clustered probing results.

(b) Bulgarian definiteness layer 7 (Standard pattern).

(c) Hindi part of speech layer 12 (G>L pattern).

(d) Russian animacy layer 2 (L>G pattern).

Figure 1: Clustering of the three different patterns (1a), and an example of each of the patterns (1b–1d). Solid lines are top-to-bottom rankings; dashed are random rankings; dotted are bottom-to-top rankings. "X by Y" means classifier X using ranking Y. Some lines are omitted for clarity.

the G>L pattern (Fig. 1c), the GAUSSIAN classifier performs exceptionally well, providing higher accuracy (after a certain point) using a random or even a bottom-to-top ranking, than the LINEAR classifier using its top-to-bottom ranking. In the L>G pattern (Fig. 1d), the GAUSSIAN classifier fails quickly, and thus the LINEAR classifier provides higher accuracy using a random or bottom-to-top ranking than the GAUSSIAN classifier using its top-to-bottom ranking. We now turn to analyze the patterns we observe.

3.2.1 Ranking methods are inherently consistent

In most configs, each classifier provides better accuracy using a top-to-bottom ranking (solid lines) compared to the bottom-to-top version of the same ranking (same color, dotted line), and the random ranking (dashed lines) is in between. This is also seen in our statistical significance tests (Appendix A.3). We conclude that even if they are not optimal, all ranking methods we consider generally rank task-informative neurons higher than non-informative ones.

3.2.2 Which classifier is better?

In the Standard and G>L patterns (Figs. 1b, 1c), GAUSSIAN achieves better accuracy than LINEAR when both of them use the same ranking (including top-to-bottom LINEAR), especially when using small sets of neurons. Our statistical significance tests (Appendix A.3) show that GAUSSIAN performs significantly better than LINEAR with 6 out of the 7 rankings we tried when using 10 neurons, with 5 rankings when using 50 neurons, and with 5 rankings when using 150 neurons. We now turn to analyze what makes GAUSSIAN more successful, and show some exceptions.

GAUSSIAN is memorizing Across all configs, LINEAR provides higher selectivity than GAUSSIAN using any ranking, after a certain point (Appendix A.4). This means that GAUSSIAN tends to memorize the word-type–label pair when solving the task. While this is apparent in all configs, we note that specifically in the part-of-speech attribute there is a large portion of function words, i.e. closed set labels (e.g., pronouns, determiners), meaning that memorization can significantly help solve the task. Thus, most configs involving part of speech belong to the G>L pattern.

LINEAR is more stable On the other hand, pattern L>G shows that there are certain configs where GAUSSIAN is struggling to model the distribution, resulting in mediocre accuracy results—which even start decreasing at some point—sometimes even below majority baseline, as seen in Fig. 1d. This has also been mentioned in Torroba Hennigen et al. [2020]. We hypothesize that it happens because in those configs, there are only a few (or no) dimensions that are informative for the attribute and are Gaussian-distributed. Thus, the GAUSSIAN classifier tries to model these distributions with the wrong tools, and fails. Poor modeling then leads to wrong predictions and low accuracy. In general, LINEAR behaves similarly across configs, making it more stable.

3.2.3 Which ranking is better?

When looking at rankings, we would like to compare performance of the same classifier, using different rankings. We would expect that each classifier would perform best when using the ranking it has generated. However, this is not always the case. As we can see in all patterns in Fig. 1, for small sets of neurons, LINEAR actually achieves better accuracy when using GAUSSIAN's ranking (solid green) than its own ranking (solid orange). As the number of neurons increases, at some point its accuracy with its own ranking becomes higher than with GAUSSIAN's ranking.

We suggest two explanations for this phenomenon: First, due to its greediness, the GAUSSIAN ranking is not guaranteed to provide the optimal subset. For a subset of size 1, it goes over all possibilities, but as the size grows there are more subsets that are not taken into consideration in the algorithm, so it is more likely to miss the best sets. Second, GAUSSIAN assumes the embedding distribution to be Gaussian. On dimensions which are not Gaussian-distributed, it makes a less accurate evaluation of the contribution of each neuron. So, if a neuron is informative towards the attribute but is not Gaussian-distributed, its addition to the selected neurons set is unlikely to improve performance, and thus it is not selected. This is a problem with a performance-based selection criterion, where the selection of neurons depends on the performance of the probe.

To summarize, it seems that GAUSSIAN is good at selecting specific informative neurons, but misses the rest. While LINEAR's ranking is not optimal (it is definitely worse then GAUSSIAN's on small sets), it does seem to be more stable on different sizes. PROBELESS provides decent performance (and is inherently consistent), but is usually behind the other two.

4 Interventions

The variance of results in our probing experiments can mostly be attributed to probing limitations [Hewitt and Liang, 2019, Pimentel et al., 2020, Belinkov, 2021, Ravichander et al., 2021], and emphasizes the need to distinguish between two properties: the probe's classification quality, and the neuron-ranking quality. To isolate the latter, and to shed light on which ranking prefers neurons that are actually being used by the model, we take a second ranking-evaluation approach: we intervene by modifying the representation in the neurons selected by the ranking, and observe how (and if) our intervention affects the language model output. This approach is more of a causal one, inspired by similar prior work [Giulianelli et al., 2018, Elazar et al., 2021, Feder et al., 2020, Vig et al., 2020, Dai et al., 2021], and can provide neuron-level explanations to some of the model's decisions. We note that in this section, we use only the ranking itself, detaching it from any probes, thus removing classification quality from ranking comparisons.

Formally, for a representation $h \in H$, ranking $\Pi(d)$ (corresponding to an attribute F) and an increasing $k \in \mathbb{N}$, we intervene by modifying h only in the $\Pi(d)_{[k]}$ neurons, and observe the effect our intervention had on the model's output—the word prediction (given the modified representation).[3] We divide the model to two components: an encoder $E : V \to H$ and a decoder $D : H \to V$, such

[3] We apply the intervention on representations of all words that possess the attribute in the sentence.

that for interventions in layer i, E is composed of all the layers of the model up to (including) i, and D is composed of all of the rest of the layers. After receiving a representation $h = E(w)$ for word w, we intervene and modify h to get a new representation h'. If $D(h) \neq D(h')$, then D is using the information we modified.

However, knowing that the information is being used is not enough; we would like to know to what purpose it is being used, and to verify that it only affects the specific attribute we are interested in. Thus, we perform a finer-grained analysis, and check if $D(h')$ is similar, to some extent, to $D(h)$. For that, we define a lemmatizer $L : V \rightarrow V$, which maps words to their lemmas, and an analyzer $A : V \rightarrow Z$, which maps words to their task labels. Our goal is to intervene such that $L(D(h)) = L(D(h'))$, but $A(D(h)) \neq A(D(h'))$. If this is the case, it implies that we have successfully identified where the task-relevant information that D uses is encoded, and how it is being used.

4.1 Intervention Methods

We consider two methods for modifying $h_{\pi(d)_k}$, and compare them.

Ablation A common modification method is trying to remove the information [Ravfogel et al., 2020] by ablating some neurons [Bau et al., 2019, Lakretz et al., 2019], meaning we set $h_{\pi(d)_k} = 0$. By that we aim to erase the information encoded in $h_{\pi(d)_k}$.

Translation For a word $w \in \mathcal{V}$ with attribute label $z \in Z$, we attempt to translate its representation (in the geometric sense) to produce a word with attribute label $z' \in Z, z \neq z'$ by taking a step in the direction of z', where bigger steps are applied to neurons that are marked as more important for the attribute. Formally, we apply the following protocol:

1. We calculate $q(z)$ and $q(z')$ as in eq. (1).
2. We set
$$h_{\Pi(d)_{[k]}} = h_{\Pi(d)_{[k]}} + \alpha_k(q(z')_{\Pi(d)_{[k]}} - q(z)_{\Pi(d)_{[k]}}) \tag{2}$$
where $\alpha \in \mathbb{R}^d$ is a log-scaled coefficients vector in the range $[0, \beta]$, such that the coefficient of the highest-ranked neuron is β and that of the lowest-ranked neuron is 0, and β is a hyperparameter.

Note that the rest of the neurons—those not in $\Pi(d)_{[k]}$—remain unaffected. Using this protocol, we give each neuron its own special treatment—an approach that was not applied before (as far we know). This can be seen as a generalization of Gonen et al. [2020].

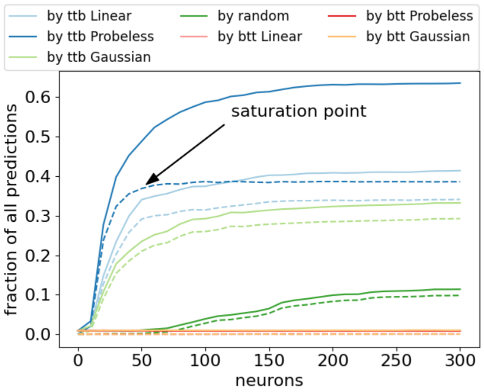
4.2 Experimental Setup

We handle the data the same way as in our probing experiments. However, since we analyze the model's predictions—which may be different from the original input—we do not have gold morphology labels anymore. Thus, for morphologically analyzing the model's predictions (L and A), we use spaCy [Honnibal et al., 2020]. Out of the languages we used in our probing experiments, in this section we are limited only to those that are supported by spaCy (English, Spanish and French). We calculate $q(z)$ based on the entire training set, and perform our interventions on the test set. We compare the same 7 rankings we used in our probing experiments (§ 3.1).

4.3 Metrics

Error Rate For our intervention experiments, we first measure the error rate of the language model. We want error rate to be high, since high error rate means we modified parts of the representation that have been used by the model in its prediction.

Correct Lemma, Wrong Value (CLWV) While inspecting predictions that are wrong after intervening ($M(h') \neq w$, where w is the true word), we categorize them by $L(M(h'))$ and $A(M(h'))$. If our intervention were successful, meaning we changed only the word's specific attribute, and not other information, then we expect to see $L(M(h)) = L(M(h'))$ and $A(M(h)) \neq A(M(h'))$. For example, if the word "makes" becomes "made" when intervening for tense, then it counts as a



(a) Spanish gender layer 2, translation results with $\beta = 8$. Solid lines are error rates, dashed are CLWVs.

	Linear	Gaussian	Probeless
English tense	$0.39, 60$ $0.37, 50$ $0.51, 60$	$0.26, 150$ $0.34, 70$ $0.41, 120$	$0.38, 30$ $0.34, 30$ $0.46, 30$
Spanish number	$0.28, 110$ $0.26, 50$ $0.23, 150$	$0.19, 100$ $0.20, 40$ $0.16, 140$	$0.35, 60$ $0.25, 30$ $0.40, 80$
Spanish gender	$0.29, 50$ $0.29, 50$ $0.26, 130$	$0.25, 80$ $0.31, 50$ $0.16, 110$	$0.37, 50$ $0.33, 30$ $0.35, 60$

(b) CLWV value at saturation point and number of neurons modified at the saturation point, using the translation method on different settings, with $\beta = 8$. In each cell, the three lines refer to layers 2, 7 and 12 respectively.

Figure 2: Translation error rate and CLWV results from a number of settings.

CLWV, but if it becomes "make" or "prepared" it does not. Thus, we define CLWV as the portion of those errors out of all predictions.

4.4 Results

4.4.1 Ablation is not Effective

Generally, across most languages, attributes and rankings we use, about 400 neurons from layer 2 and 200–300 neurons from layers 7 and 12 can be ablated without any implications on the output, meaning error rate remains the same; an example is shown in Appendix A.5. Moreover, when we do encounter more errors made by the model, our analysis does not reveal any pattern among those errors, i.e. CLWV is very low. By qualitatively analyzing those errors we see that most predicted words are mostly simply common words, e.g., "and", "if" in English. After ablating 600–700 (80%–90%) neurons from the representations, we observe a lot of errors, but most of them are because the word is predicted as nonsensical punctuation.

One major concern is that in these experiments, in some configs ablating by a bottom-to-top ranking provides better results than by the top-to-bottom version of the same ranking. In general, there are no distinct differences between the rankings. Thus, from now on we focus on translation rather than ablation.

4.4.2 Translation is Effective

Across all translation experiments (Fig. 2a shows one example), CLWV increases until a certain saturation point, after which it remains constant or drops a little [4]. This means that we reached neurons that are not relevant for the attribute, and modifying them can result in loss of other information—error rate grows while CLWV does not. Thus, we are interested in the CLWV value at the saturation point (higher is better), and in the number of neurons modified at the saturation point (lower is better). We report these values in a number of configs in Table 2b, and the rest of the configs in Appendix A.6.

Compared to ablation, translating a relatively small number of neurons results in a higher error rate, and these errors are closer to what we would expect. For example, translating only 50 neurons in Spanish gender layer 2 results in 37% CLWV and 49% error rate, while ablating 50 neurons from the same config and ranking gives no CLWV errors and only 1% of error rate. We further note that unlike in ablation experiments, here our rankings are inherently consistent, i.e. across all configs, all top-to-bottom rankings perform better than the rest, while random ranking's error rate

[4] We define "saturation point" as the first point from which there are two consecutive points where the value increase is by a factor lower than 1.05.

sometimes increases a little, and bottom-to-top rankings do not manage to affect the model's output at all (Fig. 2a).

4.4.3 PROBELESS is the Most Effective Ranking for Interventions

A clear trend from our results (Tables 2b, 2 and Fig. 2a) is that in most cases, PROBELESS achieves higher CLWV values, and does so using a smaller number of neurons, than the other two rankings. Furthermore, its error rate is significantly higher than the other two. This implies that PROBELESS tends to select neurons that are being used by the model, more so than the other rankings. However, while it does select neurons that are relevant for the attribute in question (CLWV is relatively high), it also tends to select neurons that are used by the model for other kinds of attributes (the difference between error rate and CLWV is relatively high).

Among LINEAR and GAUSSIAN, LINEAR seems to have the upper hand, with higher CLWV values in most configs. This provides another evidence that the superiority of GAUSSIAN in the probing experiments may be due to the quality of its classifier, and specifically its memorization ability, rather than the quality of the ranking it produces, as here only the ranking affects results.

5 Discussion and Conclusion

In this work, we compare different methods for ranking neurons according to their importance for a morphological attribute. We show that to evaluate a ranking in a probing scenario, one should separate between the ranking itself and the quality of the classifier that is using the ranking. We also show the need to evaluate a ranking using causal methods, such as interventions, especially for applications such as controlling or debugging the model. We propose a new ranking method that relies solely on the data, without training any auxiliary classifier, and show that it is valid, and prefers neurons that are being used by the model, more so than other ranking methods. We also propose a method for intervening within the model's representations such that it transforms the output in a desired way.

5.1 Encoded Information and Used Information are not the Same

When modifying the selected neurons, in most configs, top-to-bottom PROBELESS affects the output more than top-to-bottom GAUSSIAN and top-to-bottom LINEAR, and does so using a smaller number of neurons. This is in contrast to our probing results, where PROBELESS rarely overcame the other two, implying that encoded information and used information are not the same, and high probing accuracy does not necessarily entail that the information is actually important for the model. This finding is in line with previous work [Elazar et al., 2021, Ravichander et al., 2021].

5.2 Individual Neurons are Important

In our intervention experiments, modifying too many neurons results in more errors that are not related to the true word. This proves the importance of looking into individual neurons, especially when trying to intervene in the inner workings of the model. For example, Gonen et al. [2020] try to change the language of a word by intervening with the representation, using the same translation method we use, but with the same coefficient for every neuron, and on the entire representation. Our results imply that they may get better results by using our method.

Acknowledgments

We thank Lucas Torroba Hennigen for his helpful comments. This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 448/20) and by an Azrieli Foundation Early Career Faculty Fellowship. YB is supported by the Viterbi Fellowship in the Center for Computer Engineering at the Technion.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 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, 2017. URL https://openreview.net/forum?id=BJh6Ztuxl.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 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)*, pages 2126–2136, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://www.aclweb.org/anthology/P18-1198.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. Causalm: Causal model explanation through counterfactual language models. *Computational Linguistics*, pages 1–52, 2020.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. Identifying and controlling important neurons in neural machine translation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1z-PsR5KX.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://www.aclweb.org/anthology/P19-1580.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. Poor man's bert: Smaller and faster transformer models, 2020.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James R. Glass. What is one grain of sand in the desert? analyzing individual neurons in deep NLP models. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6309–6317. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33016309. URL https://doi.org/10.1609/aaai.v33i01.33016309.

Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. Intrinsic probing through dimension selection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 197–216, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.15. URL https://www.aclweb.org/anthology/2020.emnlp-main.15.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://www.aclweb.org/anthology/D19-1275.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus

Aranzabe, H̑órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Ethan Chi, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phùòng Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lùòng Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Arzucan Özgür, Balkız Öztürk Başaran, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Djamé Seddah,

Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Steinh͡ór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, and Anna Zhuravleva. Universal dependencies 2.7, 2020. URL http://hdl.handle.net/11234/1-3424. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Maxim Binshtok, Ronen I Brafman, Solomon Eyal Shimony, Ajay Martin, and Craig Boutilier. Computing optimal subsets. In *AAAI*, pages 1231–1236, 2007.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and alternatives. *CoRR*, abs/2102.12452, 2021. URL https://arxiv.org/abs/2102.12452.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. UniMorph 2.0: Universal Morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1293.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6011. URL https://www.aclweb.org/anthology/W18-6011.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.295.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5426. URL https://www.aclweb.org/anthology/W18-5426.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers, 2021.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1002. URL https://aclanthology.org/N19-1002.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. It's not Greek to mBERT: Inducing word-level translations from multilingual BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 45–56, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.5. URL https://www.aclweb.org/anthology/2020.blackboxnlp-1.5.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

A Appendix

A.1 Data Preparation

We remove any sentences that would have a sub-token length greater than 512, the maximum allowed for M-BERT, the language model we use for generating representations. As in Torroba Hennigen et al. [2020], we remove attribute labels that are associated with less than 100 word types in any of the data splits. This mostly removes function words, and we found it makes it harder for probes to use memorization for solving the task.

A.2 Clustering probing results

For each config out of the 156 we experimented with, we have results of 14 classifier–ranking combinations, each of length 150, the max k (number of neurons) we used. For clustering these results, we first remove all combinations involving a bottom-to-top ranking, as these add a lot of noise to the clustering algorithm, making it focus on irrelevant signals. Thus, our results matrix is of shape $[156, 8, 150]$. We then reshape the matrix to shape $[156, 8 \times 150]$ and run K-means over it with $K = 3$. Projecting the K-means output with t-SNE gives us Fig. 1a.

A.3 Statistical Significance Tests

Table 1 shows the results of our statistical significance tests. The three rows in each cell correspond to using 10, 50 and 150 neurons. If there is an * in the $[i, j]$ cell, is means that the p-value under the null hypothesis that probe j is better than probe i is lower than 0.05, when using the matching number of neurons. For example, we see that there is an * in the first and second rows in the $[0, 3]$ cell, meaning we can confidently reject the hypothesis that LINEAR by LINEAR is better than GAUSSIAN by GAUSSIAN when using 10 or 50 neurons, but we cannot do so for 150 neurons. In fact, looking at the $[3, 0]$ cell shows us that when using 150 neurons, GAUSSIAN by GAUSSIAN is not better than LINEAR by LINEAR.

While we do not show random and bottom-to-top rankings in Table 1 for clarity, we asserted that each classifier is statistically significant better when using a top-to-bottom ranking compared to a random ranking, and when using a random ranking compared to a bottom-to-top ranking.

A.4 Probing Selectivity

A selectivity example is provided in Fig. 3. In all configs, LINEAR is significantly more selective than GAUSSIAN using any ranking.

A.5 Ablation Results

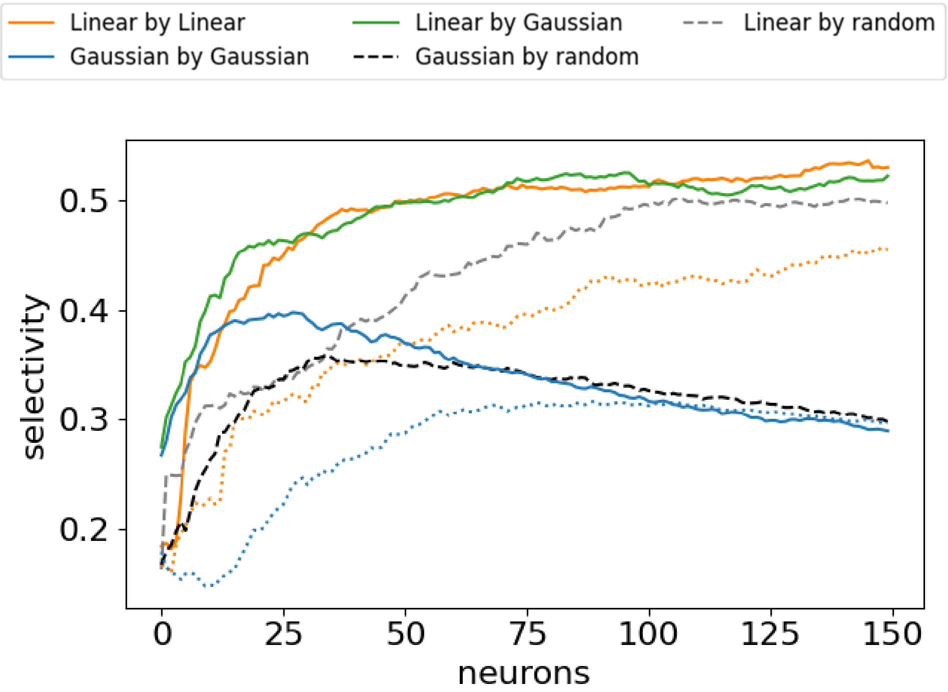
One ablation example is shown in Fig. 4. No matter the ranking, ~ 400 neurons can be ablated with little impact on the output, and CLWV remains low.

A.6 Translation results

The rest of the translation results (complementing Table 2b) is found in Table 2.

	G by ttb G	L by ttb G	G by ttb L	L by ttb L	G by ttb P	L by ttb P
G by ttb G	—	* * *	* * *	* *	* * *	* *
L by ttb G		—	*	*	* *	* *
G by ttb L		*	—	*	* *	*
L by ttb L	*	* *	*	—	* *	* *
G by ttb P			*	*	—	*
L by ttb P		*		*	*	—

Table 1: Statistical significance results. G, L, P and ttb are abbreviations for GAUSSIAN, LINEAR, PROBELESS and top-to-bottom, respectively.



(a) Hindi part of speech layer 12 selectivity.

Figure 3: Selectivity example. Solid lines are top-to-bottom rankings; dashed are random rankings; dotted are bottom-to-top rankings. Some lines are omitted for clarity.

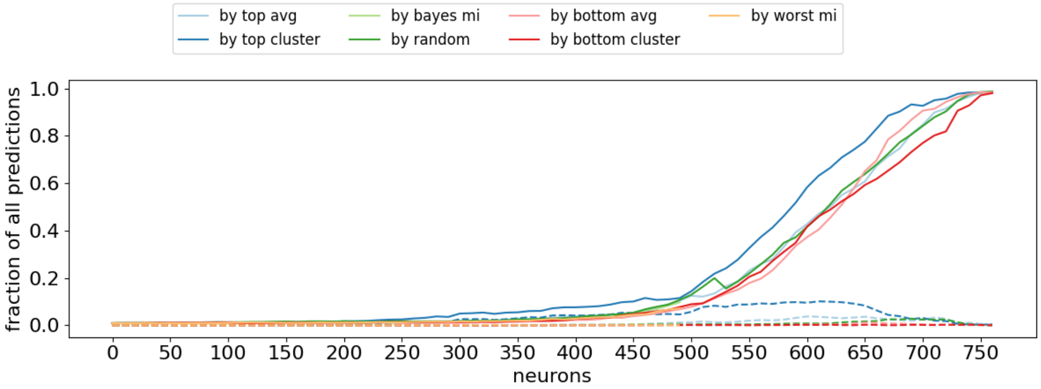


Figure 4: Spanish gender layer 2, ablation results. Solid lines are error rates, dashed are CLWVs.

	LINEAR	GAUSSIAN	PROBELESS
English number	$0.04, 70$ $0.09, 50$ $0.11, 130$	$0.02, 60$ $0.07, 30$ $0.04, 60$	$0.06, 90$ $0.11, 50$ $0.17, 110$
Spanish tense	$0.20, 110$ $0.16, 80$ $0.31, 130$	$0.15, 140$ $0.11, 70$ $0.18, 70$	$0.27, 60$ $0.20, 60$ $0.33, 60$
French number	$0.19, 110$ $0.18, 50$ $0.07, 110$	$0.09, 150$ $0.17, 30$ $0.11, 150$	$0.25, 60$ $0.20, 30$ $0.33, 120$
French tense	$0.10, 110$ $0.10, 120$ $0.14, 150$	$0.01, 90$ $0.06, 100$ $0.07, 110$	$0.13, 70$ $0.08, 70$ $0.15, 90$
French gender	$0.17, 80$ $0.16, 40$ $0.14, 170$	$0.17, 80$ $0.16, 40$ $0.06, 140$	$0.22, 60$ $0.17, 30$ $0.20, 60$

Table 2: CLWV value at saturation point and number of neurons modified at the saturation point, using the translation method on different settings. In each cell, the three lines refer to layers 2, 7 and 12 respectively.