ON THE PITFALLS OF ANALYZING INDIVIDUAL NEURONS IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

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 two pitfalls in this methodology: 1. It confounds distinct factors: probe quality and ranking quality. We separate them and draw conclusions on each. 2. It focuses on encoded information, rather than information that is used by the model. We show that these are not the same. We compare two recent ranking methods and a simple one we introduce, and evaluate them with regard to both of these aspects.[1]

1 INTRODUCTION

Many studies attempt to interpret language models by predicting different linguistic properties from word representations, an approach called probing classifiers (Adi et al., 2017; Conneau et al., 2018, *inter alia*). A growing body of work focuses on individual neurons within the representation, attempting to show in which neurons some information is encoded, and whether it is localized (concentrated in a small set of neurons) or dispersed. Such knowledge may allow us to control the model's output (Bau et al., 2019), to reduce the number of parameters in the model (Voita et al., 2019; Sajjad et al., 2020), and to gain a general scientific knowledge of the model. The common methodology is to train a probe to predict some linguistic attribute from a representation, and to use it, in different ways, to rank the neurons of the representation according to their importance for the attribute in question. The same probe is then used to predict the attribute, but using only the k-highest ranked neurons from the obtained ranking, and the probe's accuracy in this scenario is considered as a measure of the ranking's quality (Dalvi et al., 2019; Torroba Hennigen et al., 2020; Durrani et al., 2020). We see this framework as **Pitfall I**: Two distinct factors are conflated—the probe's classification quality and the quality of the ranking it produces. 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.

Another shortcoming of the current methodology, which we mark as **Pitfall II**, is the focus on encoded information, regardless of whether it is actually used by the model in its language modeling task. A few studies (Elazar et al., 2021; Feder et al., 2020) have considered this question, and shown that encoded information is not necessarily being used for language modeling, but these do not look at individual neurons. We argue that in order to evaluate a ranking, one should also examine if, and how, the k-highest ranked neurons are used by the model for the attribute in question—which would allow some control over the model's output.

In this work, we analyze three neuron ranking methods. Two 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. The third is a simple ranking method we propose, 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.

We experiment with disentangling probe quality and ranking quality, by using a probe from one method with a ranking from another method, and comparing the different probe–ranking combinations.

[1] Code for reproducing our results is publicly available at: <ANONYMIZED>

We expose the problematic nature of the current methodology (Pitfall I), by showing that in some cases, a suitable probe which is given an intentionally bad ranking, or a random one, provides higher accuracy than another which is given its allegedly optimal ranking. We find that while the GAUSSIAN method generally provides higher accuracy, its probe's selectivity (Hewitt & Liang, 2019) is lower, implying that it performs the probing task by memorizing, 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.

We then turn to analyzing which ranking selects neurons that are used by the model, by applying interventions on the representation: we modify subsets of neurons from each ranking and measure the effect on language modeling w.r.t to the property in question. We highlight the need to focus on used information (Pitfall II): even though PROBELESS does not excel in the probing scenario, it selects neurons that are used by the model, more so than the two probing-based rankings. We find that there is an overlap between encoded information and used information, but they are not the same, and argue that more attention should be given to the latter.

We primarily experiment with the M-BERT model (Devlin et al., 2019) on 9 languages and 13 morphological attributes, from the Universal Dependencies dataset (Zeman et al., 2020). We also experiment with XLM-R (Conneau et al., 2020), and find that most of our results are similar between the models, with a few differences which we discuss. Our experiments reveal the following insights:

- We show the need to separate between probing quality and ranking quality, via 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.

2 NEURON RANKINGS AND DATA

We begin by introducing some notation. Denote the word representation space as $H \subseteq \mathbb{R}^d$ and an auxiliary task as a function $F : H \rightarrow Z$, 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 $\Pi(d)_{[k]} = \{\Pi(d)_1, ..., \Pi(d)_k\}$. One may wish to find an optimal ranking $\Pi^*(d)$ such that $\forall k, \Pi^*(d)_{[k]}$ is the optimal subset with respect to F. However, finding 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 prior probing-based neuron-ranking methods, and a novel ranking we propose, based on data statistics rather than probing.

LINEAR The first method, henceforth LINEAR (named linguistic correlation analysis in Dalvi et al. 2019),[2] 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. Dalvi et al. (2019) 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 original 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] A small enhancement to the algorithm was presented in Durrani et al. (2020).

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, at every iteration. This way we 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. Torroba Hennigen et al. (2020) found that most of the tasks can be solved using a low number of neurons, but also noted that their classifier is limited due to the Gaussian distribution assumption.

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 AND MODELS

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, Finnish, French, Hindi, 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 pre-trained M-BERT and XLM-R (unless stated otherwise, all results are with M-BERT), and take word representations from layers 2, 7 and 12 of each model, to see if there are different patterns in the beginning, middle and end of the models. We end up with a total of 156 different configs (language $\times$ attribute $\times$ layer) to test for each model. 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 dimension $d = 768$. We do not mask any words throughout our work.

2.3 OVERLAPS

Before evaluating our rankings in different scenarios, we first characterize them by looking at the 100-highest ranked neurons (out of 768) from different rankings, across different configs.

Some neurons are important for an attribute across languages Since we work with multilingual models, we expect to see overlap in the selected neurons for one attribute across different languages. Fig. 1 shows that for PROBELESS this is indeed the case, as some attributes share a large number of important neurons across languages. For example, number in Spanish and number in French share 70 of their 100 most important neurons, where the expected number for overlap of two random selections of neurons is only 13 (Appendix A.2). Compared to the other two rankings (Appendix A.3), PROBELESS is the most consistent across languages, while GAUSSIAN rarely shows consistency, which may be a weakness.

Some neurons are unanimously important By looking at the overlaps between important neurons selected by different rankings for the same config, we observe that for all configs, the overlap between all three rankings surpasses the expected number (which is ~ 1.69; Appendix A.2), meaning there are neurons that are recognized by all three rankings as important.

We further see that in most cases, the greatest overlap is between LINEAR and PROBELESS. We find it reasonable, as both of them aim to select neurons that separate classes the best—one by a classifier and the other by data statistics—while GAUSSIAN takes a different approach, assuming a Gaussian distribution and selecting neurons only by performance. Examples from 8 configs are shown in Fig. 2.

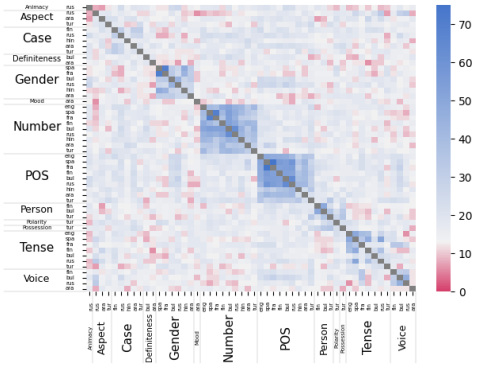


Figure 1: Layer 7 neurons overlap, using PROBELESS ranking. Blue squares are above the expected overlap between 2 rankings (Appendix A.2), red are below. Major ticks are attributes, minor are languages.

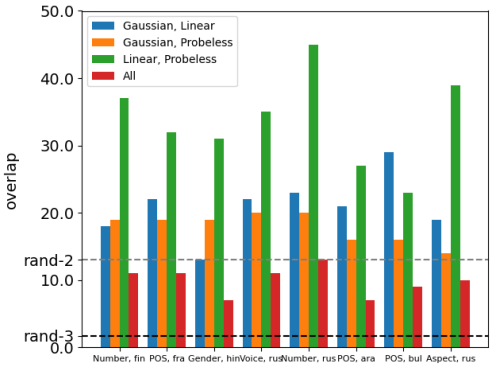


Figure 2: Layer 2 neurons overlap between every pair of rankings, from 8 randomly selected configs. Gray and black dashed lines show the expected overlap between 2 and 3 random rankings, respectively (Appendix A.2).

There are more overlaps in XLM-R Performing the same analysis across languages on XLM-R (Appendix A.3), the overlap size is at least as the expected one between all config pairs, and is usually greater. It may imply that XLM-R's representations can be pruned more easily than M-BERT's, since some neurons encode multiple attributes, and there is greater redundancy among the others.

3 PITFALL I: CLASSIFIERS VS. RANKINGS

We now turn to evaluating the rankings, and present the pitfalls in doing so. 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—Pitfall I—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 $f : H_k \to Z$ 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 neurons.[3] The assumption is that the better f performs, the more task-relevant information is encoded in $h_{\Pi(d)_{[k]}}$. Yet, the behaviour of f itself might affect results and conclusions about the ranking. Thus, we take classifier capabilities into consideration when analyzing 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 (LINEAR and GAUSSIAN). We use the hyperparameters reported in Durrani et al. (2020) and Torroba Hennigen et al. (2020) for training the classifiers. 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, referred to as top-to-bottom and bottom-to-top, respectively. To those we add a random ranking baseline, resulting in 7 different rankings overall. We compare all classifier–ranking combinations for each 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]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".

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 & Liang, 2019), 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 can memorize 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. 3a shows a t-SNE (van der Maaten & 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.4). It shows three clusters of configs, which correspond to three distinct patterns, demonstrated in Figs. 3b-3d. 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. 3b), 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 the G>L pattern (Fig. 3c), 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. 3d), 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. On XLM-R we see very similar results, and most configs follow the same pattern in each model (Appendix A.6). We now turn to analyze the patterns we observe.

3.2.1 RANKING METHODS ARE INHERENTLY CONSISTENT

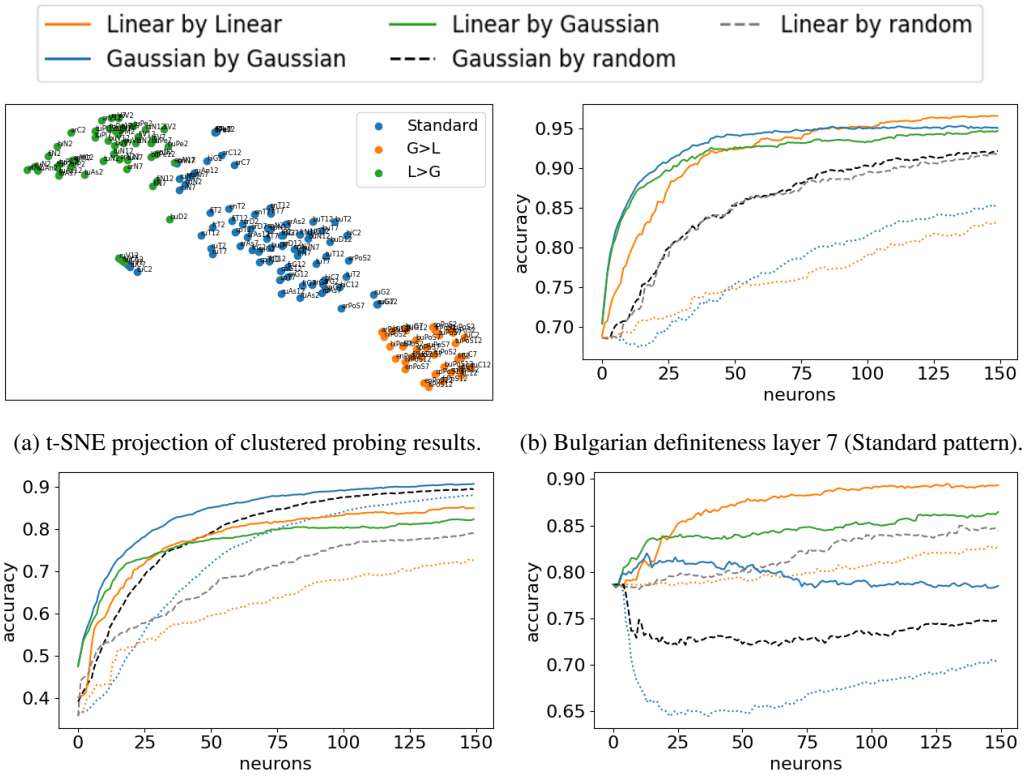
In most configs, each classifier provides better accuracy using a top-to-bottom ranking (solid lines) compared to the bottom-to-top version of the same ranking (same color, dotted line), and the random ranking (dashed lines) is in between. This is also seen in our statistical significance tests (Appendix A.5). 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. 3b, 3c), 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.5) 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.6). 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. 3d. 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



(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 3: Clustering of the three different patterns (3a), and an example of each of the patterns (3b–3d). 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.

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. 3, 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 PITFALL II: ENCODED INFORMATION VS. USED INFORMATION

The variance of results in our probing experiments can mostly be attributed to probing limitations (Hewitt & Liang, 2019; Belinkov, 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 (which is ignored by previous work—Pitfall II), we take a second ranking-evaluation approach: we intervene by modifying the representation in the neurons selected by the ranking, and observe if, and how, 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). 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).[4] For vocabulary $\mathcal{V}$, we divide the model to two components: $E : \mathcal{V} \rightarrow H$ and $D : H \rightarrow \mathcal{V}$, such that for interventions in layer i, E is composed of all of the layers of the model up to (including) i, and D is composed of all of the rest of the layers, including the classification head. After receiving a representation $h = E(w)$ for word $w \in \mathcal{V}$, 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 $L : \mathcal{V} \rightarrow \mathcal{V}$, which maps words to their lemmas, and an analyzer $A : \mathcal{V} \rightarrow Z$, which maps words to their task labels. Our goal is to intervene such that $L(D(h)) = L(D(h'))$, but $A(D(h)) \neq A(D(h'))$. For example, if we intervene for tense, we would like the word "sleeps" to become "slept". 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 by ablating some neurons (Morcos et al., 2018; 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

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

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 use only 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 ($D(h') \neq w$, where w is the true word), we categorize them by $L(D(h'))$ and $A(D(h'))$. If our intervention were successful, meaning we changed only the word's specific attribute, and not other information, then we expect to see $L(D(h)) = L(D(h'))$ and $A(D(h)) \neq A(D(h'))$; that is, correct lemma but wrong value (CLWV). For example, if the word "makes" becomes "made" when intervening for tense, then it is considered as a correct type of error, 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

Across most configs, 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.7. Moreover, when error rate does grow, CLWV is very low. By qualitatively analyzing those errors we saw that most predicted words are 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. Another major concern is that 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 here on we focus on translation rather than ablation.

4.4.2 TRANSLATION IS EFFECTIVE

Across all translation experiments (Fig. 4 shows one example, more are in Appendix A.8), CLWV increases until a certain saturation point, after which it remains constant or drops a little.[5] 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 would also like the difference between the error rate and CLWV at the saturation point to be as small as possible. All terms considered, we perform a sweep search on the values of β in the range $[1, 12]$ on a dev set. We find that low β values provide low CLWV, while high values provide higher CLWV but also widen the gap between error rate and CLWV. We find $\beta = 8$ to be a balanced point, and thus report test results with $\beta = 8$ in three configs in Table 1, and the rest of the configs in Appendix A.8. The results for XLM-R are given in Appendix A.9.

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 selected by PROBELESS in Spanish gender layer 2 results in 37% CLWV and 49% error rate, while ablating 50 neurons from the same config and ranking gives 0% CLWV errors and only 1% of error rate. We further note that unlike in ablation experiments, here our rankings are inherently consistent: across all configs, all top-to-bottom rankings perform better than the rest, while random rankings' error rate sometimes increases a little, and bottom-to-top rankings do not manage to affect the model's output at all (Fig. 4 is one example, more are in Appendix A.8).

[5] 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.

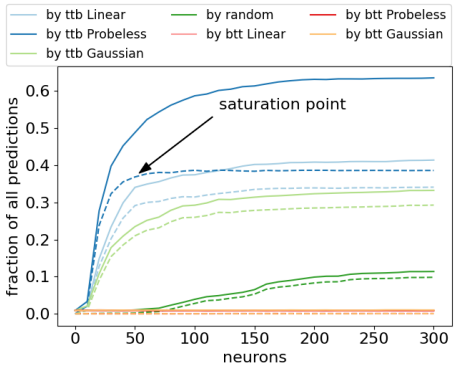


Figure 4: Spanish gender layer 2, translation results with $\beta = 8$. Solid lines are error rates, dashed are CLWVs. ttb and btt stand for top-to-bottom and bottom-to-top, respectively.

Table 1: CLWV value at saturation point and number of neurons modified at the saturation point, using the translation method on different configs, with $\beta = 8$. In each cell, the three lines refer to layers 2, 7 and 12 respectively.

	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

4.4.3 PROBELESS IS THE MOST EFFECTIVE RANKING FOR INTERVENTIONS

A clear trend from our M-BERT results (Table 1, Fig. 4 and Appendix A.8) 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. In XLM-R, it seems that LINEAR and PROBELESS both have the lead, with GAUSSIAN falling behind (Appendix A.9).

5 DISCUSSION AND CONCLUSION

In this work, we show two pitfalls with the common approach for ranking neurons according to their importance for a morphological attribute, and compare different ranking methods that follow this approach. 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—Pitfall I. While previous work concentrated on encoded information—Pitfall II—we show that it is not the same as information used by a model, by showing that GAUSSIAN is inferior in the interventions scenario, in contrast to our probing results. This implies that high probing accuracy does not necessarily entail that the information is actually important for 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.

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 finer-grained method.

Ethics statement Our work contributes to the effort of improving the interpretability of language models, and more generally of neural networks. Better explanations and controls over a model's outputs can ameliorate its fairness, for example in the case of gender bias: our intervention method can guide users on how to reduce such biases, by pointing to model components (neurons) responsible for gender and offering intervention methods to control model behavior w.r.t a particular property. On the other hand, malicious actors could use such capability to increase discrimination. Exposing the capabilities may also help develop defense mechanisms.

Reproducibility statement All our results are reproducible using the code repository we will release. All experimental details, including hyperparameters, are reported in §3.1, 4.1 and 4.2. As language models, we used the implementation of the transformers library (Wolf et al., 2020). We performed our experiments on NVIDIA RTX 2080 Ti GPU. All data preparation details are reported in §2.2 and Appendix A.1.

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.

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.

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

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

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)*, pp. 2126–2136, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://www.aclweb.org/anthology/P18-1198.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

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*, pp. 6309–6317. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33016309. URL https://doi.org/10.1609/aaai.v33i01.33016309.

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)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4865–4880, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.395. URL https://www.aclweb.org/anthology/2020.emnlp-main.395.

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*, pp. 1–52, 2020.

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*, pp. 240–248, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5426. URL https://www.aclweb.org/anthology/W18-5426.

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*, pp. 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.

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)*, pp. 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://www.aclweb.org/anthology/D19-1275.

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.

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.

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)*, pp. 11–20, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1002. URL https://aclanthology.org/N19-1002.

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)*, pp. 91–101, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6011. URL https://www.aclweb.org/anthology/W18-6011.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. Poor man's bert: Smaller and faster transformer models. *ArXiv*, abs/2004.03844, 2020.

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)*, pp. 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.

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.

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*, pp. 5797–5808, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://www.aclweb.org/anthology/P19-1580.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

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, Hórunn 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, Steinþór 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.

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 fewer 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 EXPECTED OVERLAP BETWEEN RANDOM RANKINGS

Given i rankings, we calculate the expected size of overlap between the first M neurons across all rankings:

For selecting M neurons from the range $\{1, ..., N\}$, let $C_i \in \mathbb{R}^{N \times N \times N}$ be a matrix such that in $C_i[n, m, k]$ we keep the number of possibilities to select m neurons from the range $[n]$ such that exactly k different neurons are selected by all i rankings, where $k \leq m \leq n$ and $n, m, k > 0$. For calculating $C_i[n, m, k]$ we first select the k neurons from range $[n]$ that are selected by all i rankings, thus $\binom{n}{k}$ possibilities. Then, for selecting the rest of the neurons, each ranking has to select $m - k$ neurons from the remaining $n - k$ neurons, so there are $\binom{n-k}{m-k}^i$ possibilities. From these, we want to substract the number of possibilities in which there is at least one neuron that is selected by all rankings, which is $\sum_{j=1}^{m-k} C_i[n - k, m - k, j]$. Concluding, we compute $C_i[n, m, k]$ by:

$$C_i[n, m, k] = \binom{n}{k} \left(\binom{n-k}{m-k}^i - \sum_{j=1}^{m-k} C_i[n - k, m - k, j] \right) \tag{3}$$

after initializing $C[1, 1, 1] = 1$. Then, we calculate the expected number of overlapping neurons by:

$$E_i(n, m) = \frac{\sum_{k=1}^{m} k \times C[n, m, k]}{\binom{n}{m}^i} \tag{4}$$

since $\frac{C[n,m,k]}{\binom{n}{m}^i}$ is the probability to have exactly k overlapping neurons. We thus get $E_2(768, 100) \approx 13.02$ and $E_3(768, 100) \approx 1.69$.

A.3 OVERLAPS

Fig. 5 shows overlaps between the 100 most important neurons chosen by LINEAR and GAUSSIAN for different configs. Both of them provide less overlaps then PROBELESS, with GAUSSIAN having almost no overlaps at all, showing its inconsistency across languages.

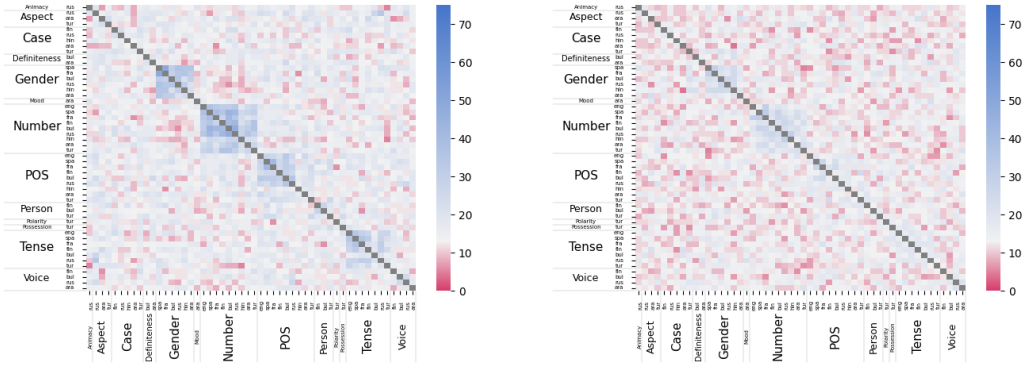
Fig. 6 presents the same analysis, but for XLM-R, with PROBELESS ranking (equivalent to Fig. 1 for M-BERT). We see far more overlaps in XLM-R, and no red squares (describing a lower overlap size than the expected one), implying that the information is more condensed in XLM-R than in M-BERT.

A.4 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 Figs. 3a and 7a.

A.5 STATISTICAL SIGNIFICANCE TESTS

Table 2 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



(a) LINEAR (b) GAUSSIAN

Figure 5: Layer 7 neurons overlap using LINEAR and GAUSSIAN rankings.

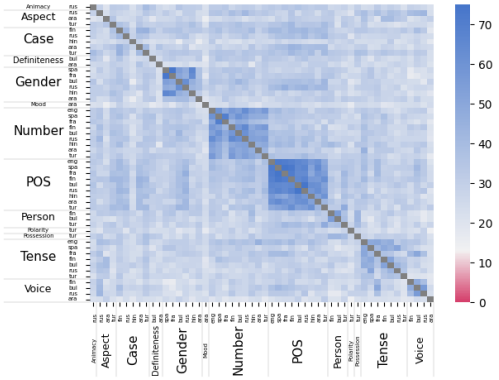


Figure 6: XLM-R layer 7 neurons overlap, using PROBELESS. Blue squares are above expected value, red are below.

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 2 for clarity, we asserted that each classifier is statistically significantly 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.6 PROBING: XLM-R AND SELECTIVITY

Fig. 7 shows XLM-R probing results. XLM-R provides very similar results to M-BERT, apparent in Figs. 7a and 7b compared to Figs. 3a and 3d, respectively.

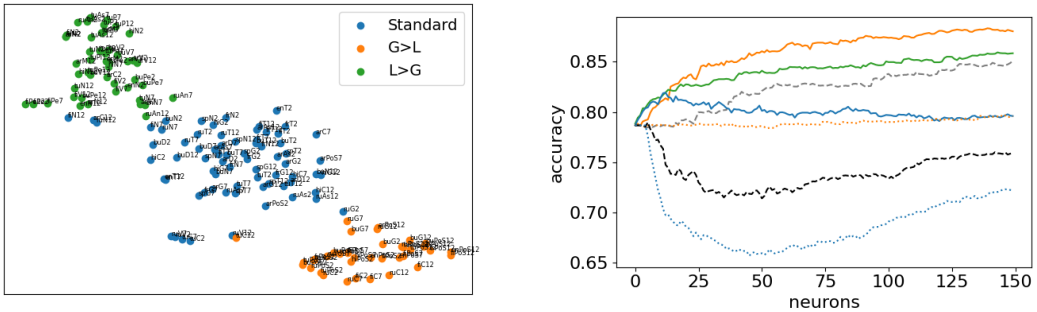
Selectivity examples from both models are provided in Fig. 8. In all configs, both in M-BERT and XLM-R, LINEAR is significantly more selective than GAUSSIAN using any ranking.

A.7 ABLATION RESULTS

One ablation example is shown in Fig. 9. No matter the ranking, ~ 400 neurons can be ablated with little impact on the output, and CLWV remains low. This behaviour is generally consistent across all configs we experimented with.

Table 2: Statistical significance results. G, L, P and ttb are abbreviations for GAUSSIAN, LINEAR, PROBELESS and top-to-bottom, respectively.

	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		*		*	*	—



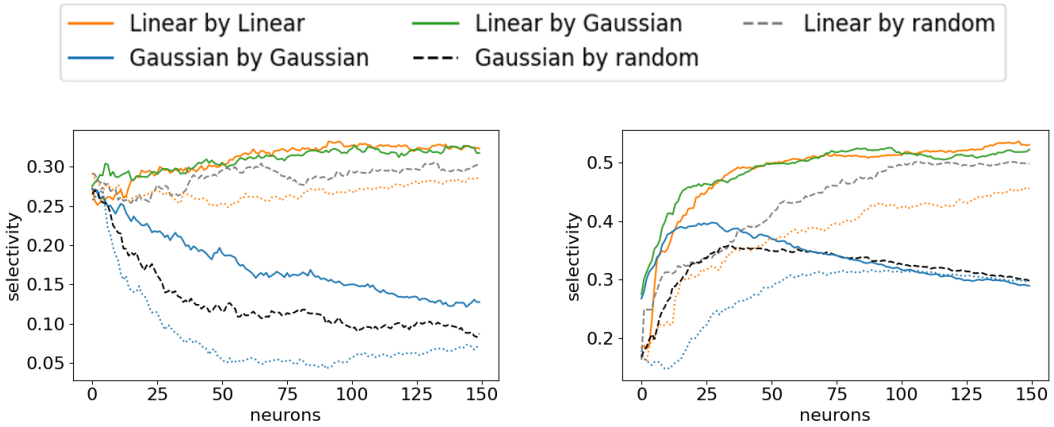
(a) t-SNE projection of clustered probing results, XLM-R.

(b) Russian animacy layer 2 accuracy, XLM-R.

Figure 7: Clustering of the three different patterns in XLM-R, and an example from one config. Solid lines are top-to-bottom rankings; dashed are random rankings; dotted are bottom-to-top rankings. Some lines are omitted for clarity.

A.8 TRANSLATION RESULTS

All of M-BERT's translation results (complementing Table 1) are found in Table 3, and examples from two configs are in Fig. 10. As described in §4.4.3, across most configs, PROBELESS achieves higher CLWV at the saturation point, and gets there earlier (using less neurons), than the other two rankings—in contrast to probing results. Among the probing-based rankings, LINEAR generally provides better results than GAUSSIAN.



(a) Russian animacy layer 2 selectivity, XLM-R.

(b) Hindi part of speech layer 12 selectivity, M-BERT.

Figure 8: Two selectivity examples from M-BERT and XLM-R. Solid lines are top-to-bottom rankings; dashed are random rankings; dotted are bottom-to-top rankings. Some lines are omitted for clarity.

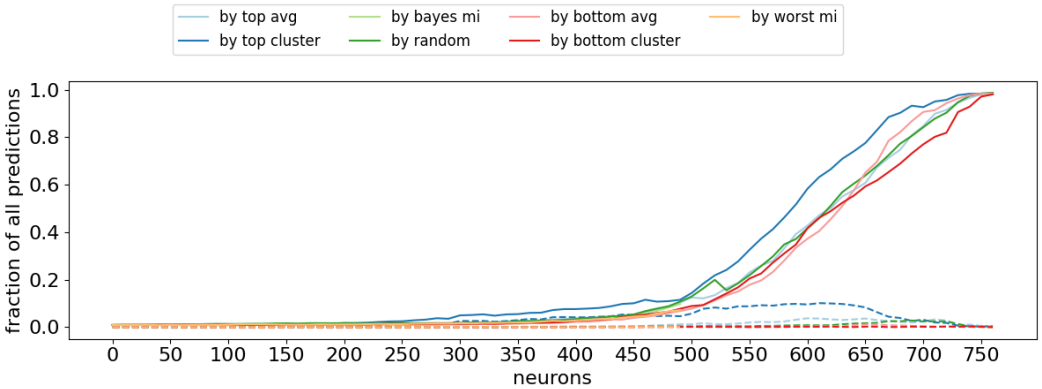
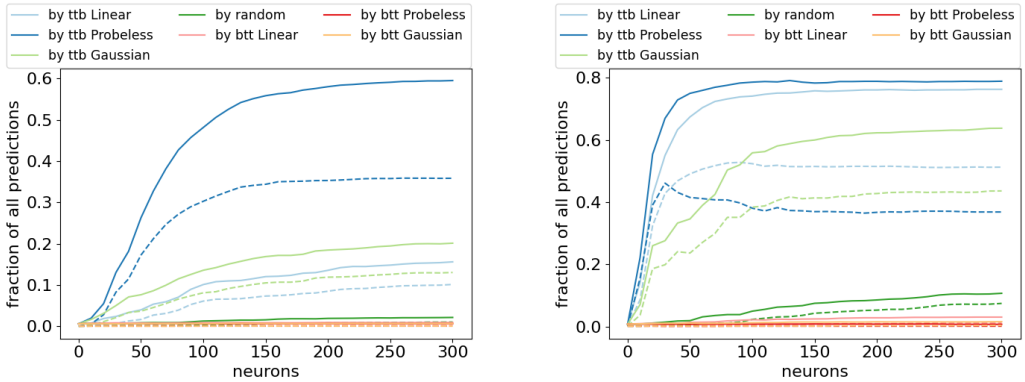


Figure 9: Spanish gender layer 2, ablation results. Solid lines are error rates, dashed are CLWVs.



(a) French number layer 12.

(b) English tense layer 12.

Figure 10: Translation results with $\beta = 8$, from two different configs. Solid lines are error rates, dashed are CLWVs. ttb and btt stand for top-to-bottom and bottom-to-top, respectively.

Table 3: CLWV value at saturation point and number of neurons modified at the saturation point, using the translation method on different configs, with $\beta = 8$. In each cell, the three lines refer to layers 2, 7 and 12 respectively.

	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
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 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
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
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

We also note that there are certain attributes that seem harder to control for, e.g., English number and French tense.

Table 4: XLM-R CLWV value at saturation point and number of neurons modified at the saturation point, using the translation method on different configs with $\beta = 8$. In each cell, the three lines refer to layers 2, 7 and 12 respectively.

	LINEAR	GAUSSIAN	PROBELESS
English number	$0.01, 130$ $0.03, 40$ $0.08, 90$	$0.00, 90$ $0.02, 30$ $0.06, 90$	$0.02, 80$ $0.04, 90$ $0.08, 90$
English tense	$0.22, 60$ $0.34, 50$ $0.35, 50$	$0.09, 190$ $0.05, 50$ $0.15, 130$	$0.21, 30$ $0.09, 60$ $0.19, 50$
Spanish number	$0.29, 70$ $0.23, 30$ $0.33, 60$	$0.11, 80$ $0.20, 50$ $0.18, 120$	$0.34, 70$ $0.23, 20$ $0.33, 40$
Spanish tense	$0.08, 90$ $0.22, 50$ $0.16, 60$	$0.00, 0$ $0.07, 70$ $0.06, 150$	$0.18, 70$ $0.19, 40$ $0.24, 80$
Spanish gender	$0.36, 60$ $0.39, 70$ $0.39, 60$	$0.17, 70$ $0.30, 30$ $0.30, 140$	$0.36, 20$ $0.33, 30$ $0.36, 20$
French number	$0.14, 110$ $0.20, 80$ $0.30, 120$	$0.06, 100$ $0.14, 60$ $0.11, 130$	$0.24, 50$ $0.27, 80$ $0.31, 70$
French tense	$0.03, 120$ $0.10, 70$ $0.10, 120$	$0.01, 120$ $0.01, 40$ $0.02, 110$	$0.07, 30$ $0.06, 20$ $0.06, 30$
French gender	$0.09, 130$ $0.16, 80$ $0.13, 90$	$0.02, 150$ $0.06, 70$ $0.05, 100$	$0.18, 50$ $0.16, 30$ $0.19, 70$

A.9 XLM-R TRANSLATION RESULTS

XLM-R translation results (equivalent to Table 3 in M-BERT) are shown in Table 4. The superiority of PROBELESS is not so clear in XLM-R compared to M-BERT, with LINEAR providing good competition. GAUSSIAN on the other hand, still falls behind.

We note that in XLM-R the CLWV values are somewhat lower compared to M-BERT. A possible explanation to that could be the difference in tokenization between the models.