What Is One Grain of Sand in the Desert?
Analyzing Individual Neurons in Deep NLP Models

Fahim Dalvi,[*1] Nadir Durrani,[*1] Hassan Sajjad,[*1]
Yonatan Belinkov,[2] Anthony Bau,[2] James Glass[2]

[1]Qatar Computing Research Institute, HBKU Research Complex, Doha 5825, Qatar
[2]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{faimaduddin,ndurrani,hsajjad}@qf.org.qa
{belinkov,abau,glass}@mit.edu

Abstract

Despite the remarkable evolution of deep neural networks in natural language processing (NLP), their interpretability remains a challenge. Previous work largely focused on what these models learn at the representation level. We break this analysis down further and study individual dimensions (neurons) in the vector representation learned by end-to-end neural models in NLP tasks. We propose two methods: *Linguistic Correlation Analysis*, based on a supervised method to extract the most relevant neurons with respect to an extrinsic task, and *Cross-model Correlation Analysis*, an unsupervised method to extract salient neurons w.r.t. the model itself. We evaluate the effectiveness of our techniques by ablating the identified neurons and reevaluating the network's performance for two tasks: neural machine translation (NMT) and neural language modeling (NLM). We further present a comprehensive analysis of neurons with the aim to address the following questions: i) how localized or distributed are different linguistic properties in the models? ii) are certain neurons exclusive to some properties and not others? iii) is the information more or less distributed in NMT vs. NLM? and iv) how important are the neurons identified through the linguistic correlation method to the overall task? Our code is publicly available[1] as part of the NeuroX toolkit (Dalvi et al. 2019).

Introduction

While neural networks have achieved state-of-the-art performance in NLP and other spheres of Artificial Intelligence (AI), their opaqueness remains a cause of concern (Doshi-Velez and Kim 2017). Interpreting the behavior of neural networks is considered important for increasing trust in AI systems, providing additional information to decision makers, and assisting ethical decision making (Lipton 2016).

Recent work attempted to analyze what linguistic information is captured in such models when they are trained on a downstream task like neural machine translation (NMT). A typical framework is to generate vector representations for some linguistic unit and predict a property of interest such as morphological features. This approach has also been applied for analyzing word and sentence embeddings (Qian,

Qiu, and Huang 2016b; Adi et al. 2016), and hidden states in NMT models (Shi, Padhi, and Knight 2016; Belinkov et al. 2017a). The analyses reveal that neural vector representations often contain substantial amount of linguistic information. Most of this work, however, targets the whole vector representation, neglecting the individual dimensions in the embeddings. In contrast, much work in computer vision investigates properties encoded in individual neurons or filters (Zeiler and Fergus 2014; Zhou et al. 2016).

We address this gap by studying individual dimensions (neurons) in the vector representations learned by end-to-end neural models. We aim to increase model transparency by identifying specific dimensions that are responsible for particular properties. We thus strive for post-hoc decomposability, in the sense of (Lipton 2016). That is, we analyze models after they have been trained, in order to uncover the importance of their individual parameters. This kind of analysis is important for improving understanding of the inner workings of neural networks. It also has potential applications in model distillation (e.g., by removing unimportant neurons), neural architecture search (by guiding the search with important neurons), and mitigating model bias (by identifying neurons responsible for sensitive attributes like gender, race or politeness[2]). In this work we lay out a methodology for identifying and analyzing individual neurons, and open the call to explore such use cases to the research community.

To this end, we propose two methods to facilitate neuron analysis. First, we perform an extrinsic correlation analysis through supervised classification on a number of linguistic properties that are deemed important for the task (for example, learning word morphology lies at the heart of modeling various NLP problems). Our classifier extracts important individual (or groups of) neurons that capture certain properties. We call this method *Linguistic Correlation Analysis*. Second, we propose an alternative methodology to search for neurons that share similar patterns in independently trained networks, based on the assumption that important properties are captured in multiple networks by individual neurons. We call this method *Cross-model Correlation Analysis*. Such an analysis is more intrinsic and help-

---

[*]Authors contributed equally
[1]https://github.com/fdalvi/NeuroX

---

[2]E.g., controlling the system to generate outputs with the right honorifics ("Sie" vs. "du") in German.

ful for highlighting important neurons for the model itself, and in the case when annotated data (supervision) may not be available. Both machine translation and language modeling are fundamental AI tasks that have seen tremendous improvements with neural networks in recent years. We evaluated our methods for analyzing neurons on these two tasks.

We provide quantitative evidence that our rankings are correct by performing several ablation experiments: from masking out important neurons to removing them completely from the training. We then conduct a comprehensive analysis of the ranked neurons. Our analysis reveals interesting findings such as i) open class categories such as *verb* (part-of-speech tag) and *location* (semantic entity) are much more distributed across the network compared to closed class categories such as *coordinating conjunction* (e.g., "but/and") or a *determiner* (e.g., "the"), ii) the model recognizes a hierarchy of linguistic properties and distributes neurons based on it, and iii) important neurons extracted from the *Cross-model Correlation* method overlap with those extracted from the *Linguistic Correlation* method; for example, both methods identified the same neurons capturing position as salient. In summary, we make the following contributions:

- A general methodology for identifying linguistically-meaningful neurons in deep NLP models.

- An unsupervised method for finding important neurons in neural networks, and a quantitative evaluation of the retrieved neurons.

- Application to various test cases, investigating core language properties through part-of-speech (POS), morphological, and semantic tagging.

- An analysis of distributed vs. focused information in NMT and NLM models.

Related Work

Much of the previous work has looked into neural models from the perspective of what they learn about various language properties. This includes analyzing word and sentence embeddings (Adi et al. 2016; Qian, Qiu, and Huang 2016b; Conneau et al. 2018), recurrent neural network (RNN) states (Shi, Padhi, and Knight 2016; Wang, Chung, and Lee 2017), and NMT representations (Belinkov et al. 2017a; 2017b; Dalvi et al. 2017). The language properties mainly analyzed are morphological (Qian, Qiu, and Huang 2016b; Vylomova et al. 2016), semantic (Qian, Qiu, and Huang 2016b) and syntactic (Shi, Padhi, and Knight 2016; Linzen, Dupoux, and Goldberg 2016; Conneau et al. 2018).

Most of this work used an extrinsic supervised task and target entire vector representations. We study the individual neurons in the vector representation and propose a simple supervised method to analyze individual/groups of neurons with respect to various properties and linguistic tasks. As an alternative to supervision which is limited to labeled data, we propose an unsupervised method based on correlation between several networks to identify salient neurons.

Some recent work on neural language models and machine translation analyzes specific neurons of length (Qian,

Qiu, and Huang 2016a; Shi, Knight, and Yuret 2016) and sentiment (Radford, Jozefowicz, and Sutskever 2017). However, not much work has been done along these lines. We present both intrinsic and extrinsic methods to analyze models at the neuron level to gain a deeper insight.

In computer vision, there has been much work on visualizing and analyzing individual units such as filters in convolutional neural networks (Zeiler and Fergus 2014; Zhou et al. 2016, among others). Even though some doubts were cast on the importance of individual units (Morcos et al. 2018), recent work stressed their contribution to predicting specific object classes via ablation studies similar to the ones we conduct (Zhou et al. 2018).

Methodology

Let $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote a sequence of input features and consider a neural network model $\mathbb{M}$ that maps $\mathbf{x}$ to a sequence of latent representations: $\mathbf{x} \xmapsto{\mathbb{M}} \mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$, where $\mathbf{z}_i \in \mathbb{R}^D$. For example, in an NMT system, $\mathbb{M}$ could be the *encoder*, $\mathbf{x}$ the input word embeddings, and $\mathbf{z}$ the hidden states. Our goal is to study individual neurons in the model $\mathbb{M}$, which we define as dimensions in the latent representation. We will use $\mathbf{z}_{ij}$ to denote the j-th dimension of the latent representation of the i-th word $\mathbf{z}_i$. We first explain a *Linguistic Correlation Analysis* method to find neurons specific to a task. Then we present a *Cross-model Correlation Analysis* method for ranking based on the correlations between neurons from different networks.

Linguistic Correlation Analysis

Consider a classification task where the goal is to predict a property $\mathbf{l}$ in a property set $\mathcal{P}$[3] that we believe is intrinsically learned in the model $\mathbb{M}$, for example word-structure (morphology) or semantic information in an NMT model. Our goal is to identify neurons in $\mathbb{M}$ that are salient for the property $\mathbf{l} \in \mathcal{P}$ being considered. We assume that we have supervision for the task in the form of labeled examples $\{\mathbf{x}_i, \mathbf{l}_i\}$ where $\mathbf{x}_i$ is the i-th word, having a property $\mathbf{l}_i \in \mathcal{P}$. Given this labeled training data, we first extract neuron activations $\mathbf{z}_i$ from the model $\mathbb{M}$ for every input word $\mathbf{x}_i$. For instance, this may be done by running the NMT encoder on the sentence and recording neuron activations for each word.

We then train a logistic regression classifier on the $\{\mathbf{z}_i, \mathbf{l}_i\}$ pairs using the cross-entropy loss. We opt to train a linear model because of its explanability; the learned weights can be queried directly to get a measure of the importance of each neuron in $\mathbf{z}_i$. From a performance point of view, earlier work has also shown that non-linear models present similar trends as of linear models in analyzing representations of neural models (Qian, Qiu, and Huang 2016b; Belinkov et al. 2017a). In order to increase interpretability and to encourage feature ranking in the classification process, we use elastic net regularization (Zou and Hastie 2005) as an additional

---

[3]A property could be a part-of-speech tag such as `verb`, or a semantic entity such as `event`, or the position of a word in a sentence. A set of properties combined constitutes a task such as POS or semantic tagging.
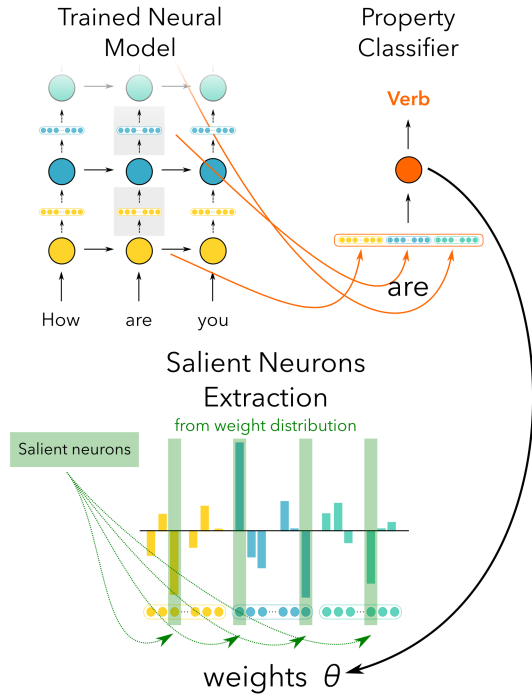
Figure 1: Linguistic Correlation Analysis: Extract neuron activations from a trained model, train a classifier and use weights of the classifier to extract salient neurons.

loss term. Formally, the model is trained by minimizing the following loss function:

$$\mathcal{L}(\theta) = -\sum_i \log P_\theta(\mathbf{l}_i|\mathbf{x}_i) + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta\|_2^2$$

where $P_\theta(l|\mathbf{x}_i) = \frac{\exp(\theta_l \cdot \mathbf{z}_i)}{\sum_{l'} \exp(\theta_{l'} \cdot \mathbf{z}_i)}$ is the probability that word i is assigned label l. The weights $\theta \in \mathbb{R}^{D \times L}$ are learned with gradient descent. Here D is the dimensionality of the latent representations $\mathbf{z}_i$ and L is the size of the label set for $\mathcal{P}$. The overall process is illustrated in Figure 1.

Elastic net regularization enjoys the sparsity effect as in Lasso regularization, which helps identify important individual neurons. At the same time, it takes groups of highly correlated features into account similar to Ridge regularization, avoiding the selection of only one feature as in Lasso regularization. This strikes a good balance between localization and distributivity. This is particularly useful in the case of analyzing neural networks where we hypothesize that the network consists of both individual focused neurons and a group of distributed neurons, depending on the property being learned. The regularization terms are controlled by hyper-parameters λ_1 and λ_2. We search for the best hyper-parameter values that maintain good accuracy while accomplishing the desired goal of selecting the salient neurons for a property, as described in the evaluation section.

Ranking Neurons: Given the trained weights of the classifier $\theta \in \mathbb{R}^{D \times L}$, we want to extract a ranking of the D neu-

---

Algorithm 1 Neuron Ranking Extraction Algorithm
1: $ordering \leftarrow []$ ▷ $ordering$ will store the neurons in order of decreasing importance
2: **for** $p = 1$ **to** 100 **by** α **do** ▷ p is the percentage of the weight mass. We start with a very small value and incrementally move towards 100%.
3:     $tnpt \leftarrow$ GETTOPNEURONSPERTAG(θ, p) ▷ $tnpt$ contains the top neurons per tag using the threshold p
4:     $topNeurons \leftarrow \bigcup_{i=1}^{L} tnpt_i$
5:     $newNeurons \leftarrow topNeurons \setminus ordering$
6:     $ordering.append(newNeurons)$
7: **end for**
8: **return** $ordering$

---

rons in the model $\mathbb{M}$. For the label of interest $\mathbf{l} \in \mathcal{P}$, we sort the weights $\theta_{\mathbf{l}} \in \mathbb{R}^D$ by their absolute values in descending order. Hence the neuron with the highest corresponding absolute weight in $\theta_{\mathbf{l}}$ appears at the top of our ranking. We consider the top n neurons (for the individual property under consideration) that cumulatively contribute to some percentage of the total weight mass as *salient neurons*. To extract a ranking of neurons w.r.t. all of the labels in $\mathcal{P}$, we use an iterative process described in Algorithm 1. We start with a small percentage of the total weight mass and choose the most salient neurons for each label $\mathbf{l}$, and increase this % iteratively, adding newly discovered top neurons to our ordering. Hence, the salient neurons for each label $\mathbf{l}$ will appear at the top of the ordering. The order in which the neurons are discovered indicates their importance to the property set $\mathcal{P}$.

Cross-model Correlation Analysis
The linguistic correlation analysis is useful for analyzing neurons given a certain property. Now, we present our Cross-model correlation method to identify neurons salient to the model $\mathbb{M}$ independent of any property. In essence, it ranks neurons according to their importance to the task the model $\mathbb{M}$ is trained on. We hypothesize that salient neurons contain important information about the task and are shared across several models. To prove this, we train multiple models $\mathbb{M}_1, \ldots, \mathbb{M}_N$ for the same task, using identical model settings but with differing training data and initialization. We then rank neurons in one of the models $\mathbb{M}_i$ by their best correlation coefficient with any neuron from a different model:

$$score(\mathbb{M}_{ij}) = \max_{\substack{1 \leq i' \leq N \\ 1 \leq j' \leq D \\ i \neq i'}} \rho(\mathbb{M}_{ij}, \mathbb{M}_{i'j'})$$

where $\mathbb{M}_{ij}$ is the j-th neuron in the i-th model and $\rho(\mathbb{M}_{ij}, \mathbb{M}_{i'j'})$ is the Pearson correlation coefficient.[4] We then consider the top neurons in this ranking as the most salient neurons for the overall model.

Evaluation using Neuron Ablation
Given the list of neurons from a trained model $\mathbb{M}$, we evaluate the rankings by challenging their presence in the net-

---

[4]Here $\mathbb{M}_{ij} \in \mathbb{R}^T$, corresponding to activations of neuron j in model i, over an evaluation set of size T words.

	French		English		German	
	POS	Morph	POS	SEM	POS	Morph
MAJ	92.8	89.5	91.6	84.2	89.3	83.7
NMT	93.2	88.0	93.5	90.1	93.6	87.3
NLM	92.4	90.1	92.9	86.0	92.3	86.5

Table 1: Classifier accuracy when trained on activations of NMT and NLM models. MAJ: local majority baseline.

Lambda 1 ↓ \ Lambda 2 →	0	1e-7	1e-6	1e-5	1e-4	1e-3	1e-2	1e-1	1
0	94.01%	94.01%	94.01%	94.01%	93.99%	93.71%	91.28%	80.51%	14.30%
1e-7	94.01%	94.01%	94.01%	94.00%	93.98%	93.70%	91.28%	80.51%	14.30%
1e-6	93.95%	93.96%	93.95%	93.95%	93.93%	93.62%	91.25%	80.50%	14.30%
1e-5	93.41%	93.40%	93.40%	93.40%	93.37%	93.09%	90.99%	80.31%	14.30%
1e-4	90.87%	90.87%	90.87%	90.87%	90.85%	90.66%	89.01%	78.81%	14.30%
1e-3	82.80%	82.80%	82.79%	82.80%	82.79%	82.71%	81.80%	67.00%	14.30%
1e-2	47.73%	47.71%	47.72%	47.73%	47.71%	47.68%	46.06%	23.51%	14.30%
1e-1	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%
1	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%	14.30%

Legend: min ▭ max

Figure 2: Effect of various values of regularization on the overall accuracy of the English POS tagging task.

work. We clamp the value of a subset of neurons to zero as in (Morcos et al. 2018) and observe the degradation in performance, reflecting how much the network is dependent on them. Our hypothesis is that an ablation of the most important neurons should cause a larger drop in performance compared to the least important neurons. We apply ablation to both the classifier (to evaluate property-specific rankings) and the original model $\mathbb{M}$ (to evaluate model-level rankings).

Ablation in Classification   Given a trained classification model, we keep N% top or bottom neurons and set the activation values of all other neurons to zero in the test set. We then reevaluate the performance of the already trained classifier. We expect to see low performance (prediction accuracy) when using only the bottom neurons versus using only the top neurons. We also retrain the classifier with only the selected N% neurons. This serves multiple purposes: i) it confirms the results from the zeroing-out method, ii) it shows that much of the performance can be regained using the selected neurons, and iii) it facilitates the analysis of how distributed a particular property is across the network.

Ablation in Neural Model $\mathbb{M}$:   Here, we want to evaluate our rankings of neurons with respect to the model $\mathbb{M}$. Given a ranked list of neurons, we incrementally zero-out N% of the neurons starting from top or bottom and report the drop in performance in terms of BLEU scores (for NMT) or perplexity (for NLM).

Experimental Settings

Neural Models: We experimented with two architectures: NMT based on sequence-to-sequence learning with attention (Bahdanau, Cho, and Bengio 2014) and an LSTM based NLM (Hochreiter and Schmidhuber 1997).[5] We trained a 2-layer bidirectional NMT model with 500-dimensional word embeddings and LSTM states. The system is trained for 20 epochs, and the model with the best development loss is used for the experiments. We follow similar settings to train a unidirectional NLM model.

Data: We experimented with English↔French (EN↔FR) and German→English (DE→EN) language pairs. We used a
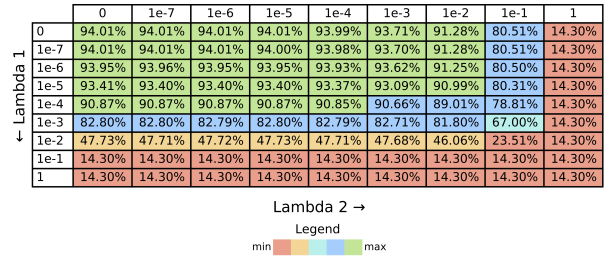
subset of 2 million sentences from the United Nations multi-parallel corpus (Ziemski, Junczys-Dowmunt, and Pouliquen 2016) for EN↔FR and from the data made available for the IWSLT campaign (Cettolo et al. 2014) for DE→EN. We split the parallel data for each language pair into three equal subsets to train three different models. For language models, we used the source side of the parallel corpora.

Language Properties: We evaluated our linguistic correlation method by selecting standard tasks of part-of-speech (POS), morphological and semantic tagging. The former two capture word structure in a language and the latter captures its nuanced meaning. Additionally we considered some general properties, such as the position of words in a sentence and predicting a *months of year* tag.

Classifier Data: We used 20k source-side sentences, randomly extracted from the MT training data, for training the classifier, and 4k sentences in the official test sets for testing. We tagged these sentences with standard taggers for the different properties; the details of these taggers can be found in the supplementary material.

Evaluation
In this section, we present the evaluation of our techniques:

Linguistic Correlation Analysis
Classifier Performance: We first evaluate the classifier performance to ensure that the learned weights are actually meaningful for further analysis and ranking extraction. The classifiers were trained using the activations of already trained neural models (NLM and NMT encoder[6]). Table 1 shows accuracy of the classifiers trained for different language pairs and tasks on a blind test set. The classifiers achieve higher accuracies compared to the local majority baseline[7] (MAJ) in all cases, except for French (POS:NLM). The overall accuracy trend shows that the neurons possess sufficient information to predict these language properties.

Since we are using elastic net regularization, we need to tune the values for λ_1 and λ_2. The regularization controls the

---

[5]We focus on standard architectures for these tasks and leave exploration of recent variants such as the Transformer (Vaswani et al. 2017) or QRNN (Bradbury et al. 2017) for future work.

[6]We limit ourselves to encoder activations for simplicity.

[7]Selecting the most frequent tag for each word and the most frequent global tag for the unknown words.

			Masking-out					
Task		ALL	10%		15%		20%	
			Top	Bot	Top	Bot	Top	Bot
NMT	FR (POS)	93.2	63.2	23.8	73.0	24.8	79.4	24.9
	EN (POS)	93.5	69.8	15.8	78.3	17.9	84.1	21.5
	EN (SEM)	90.1	51.5	16.3	65.3	18.9	74.2	20.7
	DE (POS)	93.6	65.9	15.7	78.0	15.6	88.2	15.7
NLM	FR (POS)	92.4	41.6	23.8	53.6	23.8	59.6	24.0
	EN (POS)	92.9	54.2	18.4	66.1	20.4	72.4	24.7
	EN (SEM)	86.0	49.7	21.9	56.8	22.3	65.2	25.1
	DE (POS)	92.3	39.7	16.7	51.7	16.7	67.2	16.9

Table 2: Classification accuracy on different tasks using all neurons (ALL). Masking-out: all except top/bottom N% of neurons are masked when testing the trained classifier.

(a) English Verb (#1902)

(b) German Article (#590)

(c) Position Neuron (#1903)

Figure 3: Activations of top neurons for specific properties

final ranking of the neurons directly: an increase in the value of λ_1 introduces further sparsity whereas higher values of λ_2 encourage selection of groups of correlated neurons. Our aim is to find a balance between selecting individual neurons and a group of neurons while maintaining the original accuracy of the classifier without any regularization (λ_1, λ_2 = 0). Figure 2 presents the results of a grid search over various regularization values on the English POS tagging task. The accuracy difference is minimal for λ values under $1e^{-4}$. We selected a value of $1e^{-5}$ for both λ_1 and λ_2 and used the same for all the experiments.

Neuron Ablation in the Classifier: After training the classifier, we used Algorithm 1 to extract a ranked list of neurons with respect to each property set and ablated neurons in the classifier to verify rankings. We *masked-out* all the activations (in the test set) except for the selected $N\%$ neurons and recomputed test accuracies. Table 2 summarizes the results.[8] Compared to ALL, the classification accuracy drops drastically for both NMT and NLM. However, the performance is distinctly better in the case of keeping the top N% neurons when compared to the bottom N% neurons, showing that the ranking produced by the classifier is correct for the task at-hand.

---

[8] Similar trends were found in the morphological tagging results. Please see supplementary material if interested.

Neuron	Top 10 words
#1925 (Month)	August, July, January, September, October, presidential, April, May, February, December
#1960 (Negation)	no, No, not, nothing, nor, neither, or, none, whether, appeal
#1590 (Cardinality)	50, 10, 51, 61, 47, 37, 48, 33, 43, 49

Table 3: Ranked list of words for some individual neurons in the EN-FR model.

Visualizations: have been used effectively to gain qualitative insights on analyzing neural networks (Karpathy, Johnson, and Fei-Fei 2015; Kádár, Chrupała, and Alishahi 2016). We used an in-house visualization tool (Dalvi et al. 2019) for qualitative evaluation of our rankings. Figure 3 visualizes the activations of the top neurons for a few properties. It shows how single neurons can focus on very specific linguistic properties like *verb* or *article*. Neuron #1902 focuses on two types of verbs (3rd person singular present-tense and past-tense) where it activates with a high positive value for the former ("Supports") and high negative value for the latter ("misappropriated"). In the second example, the neuron is focused on German articles. Although our results are focused on linguistic tasks, the methodology is general for any property for which supervision can be created by labeling the data. For instance, we trained a classifier to predict *position of the word*, i.e., identify if a given word is at the beginning, middle, or end of the sentence. As shown in Figure 3(a), the top neuron identified by this classifier activates with high negative value at the beginning (red), moves to zero in the middle (white), and gets a high positive value at the end of the sentence (blue). Another way to visualize is to look at the top words that activate a given neuron. Table 3 shows a few examples of neurons with their respective top 10 words. Neuron #1925 is focused on the name of months. Neuron #1960 is learning negation and Neuron #1590 activates when a word is a number. These word lists give us quick insights into the property the neuron has learned to focus on, and allows us to interpret arbitrary neurons in a given network.

Cross-model Correlation Analysis

The Cross-model correlation analysis method ranks the list of neurons based on correlation among several models. In the following, we evaluate the rankings produced by the method by ablating the neurons in the original model $\mathbb{M}$.

Neuron Ablation in Model $\mathbb{M}$: We incrementally ablate top/bottom neurons from the ranking and report the drop in performance of the NMT model. Figure 4 shows the effect of ablation on translation quality (BLEU). For all languages, ablating neurons from top to bottom (solid curves) causes a significant early drop in performance compared to ablating neurons in the reverse order (dotted curves). This validates the ranking identified by our method. Ablating just the top 50 neurons (2.5%) leads to drops of 15-20 BLEU points, while the bottom 50 neurons hurt the performance by only
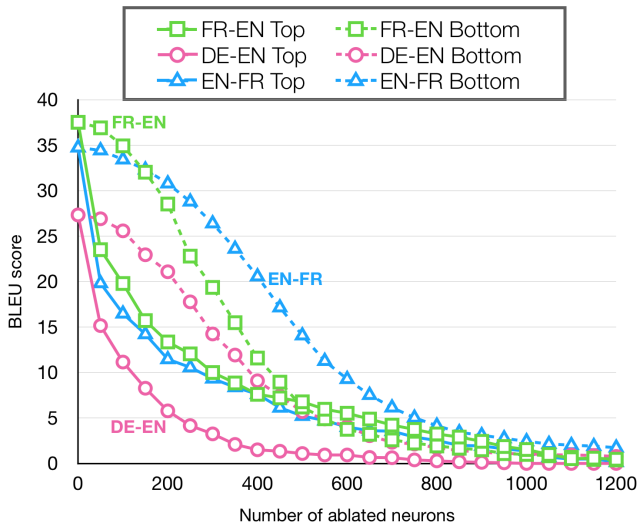
Figure 4: Effect of neuron ablation on translation performance (BLEU) when removing the top or bottom neurons based on Cross-Correlation analysis ordering.

0.5 BLEU points.

Neuron ablation in NLM: Figure 5 presents the results of ablating neurons of NLM in the order defined by the Cross-model Correlation Analysis method. The trend found in the NMT results is also observed here, i.e. the increase in perplexity (degradation in language model quality) is significantly higher when erasing the top neurons (solid lines) as compared to when ablating the bottom neurons (dotted lines).
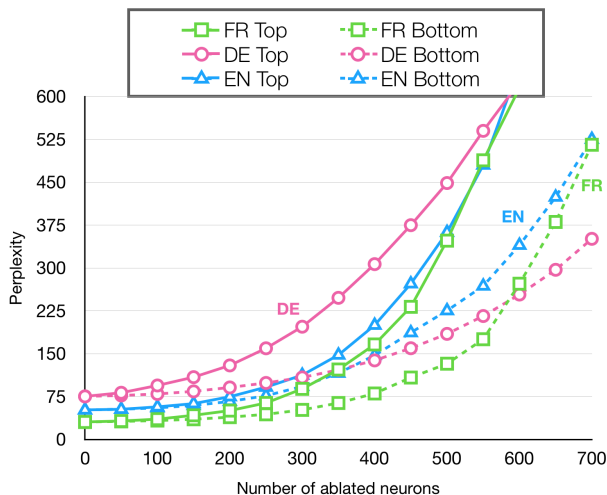

Figure 5: Effect of neuron ablation on perplexity when erasing from the top and bottom of the Cross-correlation ordering from the NLM

Extracting Neurons based on a Single Model: Recall that our Cross-model method requires multiple instances of the model to extract neuron rankings. In an effort to probe whether one instance of the model can sufficiently extract similar rankings, we tried several methods that ranked neurons of an individual model based on i) variance, and ii) distance from mean (high to low), and compared these with the ranking produced by our method. We found less than 10% overlap among the top 50 neurons of the Cross-model ranking and the single model rankings. On ablating the neurons based on several ranking methods, we found the NMT models to be most sensitive to the Cross-model ranking. Less damage was done when neurons were ablated using rankings based on variance and distance from mean in both directions, high-to-low and low-to-high (See Figure 7). This supports our claim that the Cross-model ranking identifies the most salient neurons of the model.

Comparison with Linguistic Correlation Method: Are the neurons discovered by the linguistic correlation method important for the actual model as well? Figure 8 shows the effect on translation when ablating neurons in ranking order determined by English POS and semantic (SEM) tagging, as well as top/bottom Cross-model orderings. As expected, the linguistic correlation rankings are limited to the auxiliary task and may not result in the most salient neurons for the actual task (machine translation in this case); ablating according to task-specific ordering hurts less than ablating by (top-to-bottom) Cross-model ordering. However, in both cases, degradation in translation quality is worse than ablating by bottom-to-top Cross-model ordering. Comparing SEM with POS, it turns out that NMT is slightly more sensitive to neurons focused on semantics than POS.

Analysis and Discussion

The rankings produced by the linguistic correlation and cross-correlation analysis methods give a sense of the most important neurons for an auxiliary task or the overall model. We now dive into neuron analysis based on these rankings.

Focused versus Distributed Neurons: Recall that our linguistic-correlation method provides an overall ranking w.r.t. a property set (POS/SEM tagging), and also for each individual property as described in the *Methodology* section. Here, we look at the number of salient neurons (extracted from the NMT models) for several different linguistic properties,[9] as shown in Figure 6. For example, in open-class categories such as nouns (NN/NOM), verbs (VB/VER.simp/VVPP) and adjectives (JJ/ADJ), the information is distributed across several dozen neurons. In comparison, categories such as end of sentence marker (SENT) or WH-Adverbs (WRB) and post-positions (APPO in German) required fewer than 10 neurons. We observed similar trend in the semantic tags: information about closed-class categories such as months of year (MOY) is localized in

---

[9]We choose salient neurons for each label by selecting the top neurons that cumulatively represent 25% of the total weight mass.
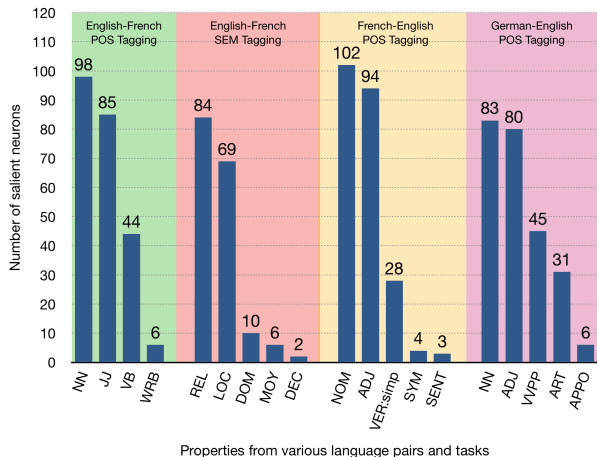
Figure 6: focused versus distributed tags: NN/NOM = Noun, JJ/ADJ = Adjective, VB = Verb, WRB = WH-Adverb, REL = relation, LOC = Location, DOM = Day of Month, MOY = Month of Year, DEC = Decade, VER:simp = Verb simple past, SENT = Full stop, VVPP = Participle Perfect, ART = Article, APPO = Post-position
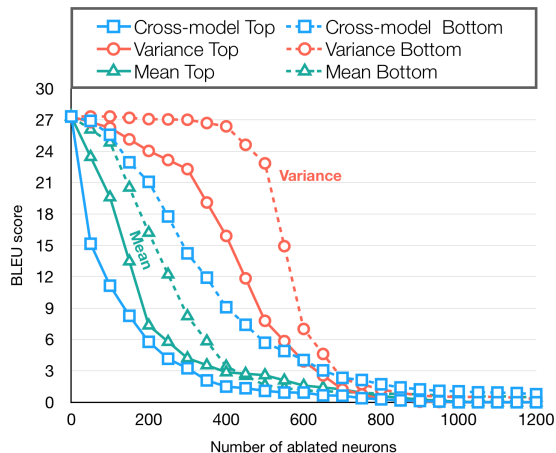

Figure 7: Cross-model ranking compared with single model statistics in DE-EN model. Variance is the ranking based on high variance to low variance. Mean is the ranking from high to low distance from mean.
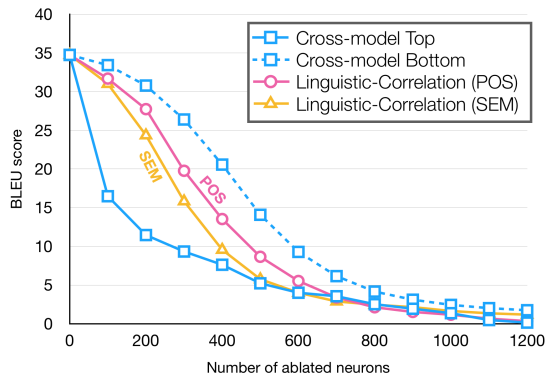

Figure 8: Effect on translation when ablating neurons in the order determined by both methods on the EN-FR model

just a couple of neurons. In contrast, an open category like location (`LOC`) is very distributed.

Shared Neurons within and across Properties: Since some information is distributed across the network, we expect to see some neurons that are common across various properties, and others that are unique to certain properties. To investigate this, we intersect top ranked neurons coming from two different properties. Some of these comparisons are interesting. For instance, we found some common neurons across all forms of adjectives, but some neurons specifically designated to specialized adjectives (e.g., comparative (`JJR`) and superlative (`JJS`) adjectives). Similarly across tasks (POS vs. Morph), we found multiple neurons targeting different verb forms (`V--F3s` and `V--F3p`, Verb Future 3rd person singular and plural) in the fine-grained morphological tagging that are aligned with a single neuron targeting the future tense verb tag (`VER:futu`) in POS tagging. This demonstrates that model recognizes a hierarchy of linguistic properties and distributes neurons based on it.

Retraining Classifier with the Selected Neurons: In the evaluation section for our linguistic-correlation classifier, we masked-out a majority of the neurons and compared the accuracy trends to confirm our ranking. An alternative to analyze is to retrain the classifier with the top or bottom N% neurons alone. Table 4 shows the results after retraining. There are several points to note here: i) training the classifier using top neurons performs consistently better than using bottom neurons, reinforcing our previous finding. ii) The classifier is able to regain performance substantially (compared to `ALL`), even using only 10% neurons. iii) Using the bottom N% neurons also restores performance (although not

as much as using the top neurons). This shows that the information is distributed across neurons. However, the distribution is not uniform, which results in a large difference between training using top and bottom neurons (i.e., the information distribution is skewed towards the top neurons as expected). Notably, using only 20% of the top neurons, the classifier is able to regain much of the performance drop in most of the cases. This finding entails that our method could be useful for model distillation purposes.

Cross-model Correlation Ranking: Analyzing the top neurons identified by our Cross-model correlation method, we found several neurons corresponding to the position of the word in a sentence. Word position has been previously found to be an important property in NMT (Shi, Knight, and Yuret 2016). The fact that our method ranks position neurons among the top ranking neurons shows its efficacy. We also observed that the top position neurons identified by our

		Re-training					
Task	ALL	10%		15%		20%	
		Top	Bot	Top	Bot	Top	Bot
NMT FR (POS)	93.2	88.4	72.1	90.0	77.8	91.1	81.8
EN (POS)	93.5	89.1	80.6	90.5	84.8	91.2	87.2
EN (SEM)	90.1	85.6	73.4	87.0	77.8	87.8	80.8
DE (POS)	93.6	91.4	77.1	92.3	81.9	92.8	85.3
NLM FR (POS)	92.4	83.7	61.8	86.2	71.7	87.8	77.4
EN (POS)	92.9	85.8	62.4	88.2	72.5	89.4	79.2
EN (SEM)	86.0	78.9	67.8	81.4	74.1	82.7	77.6
DE (POS)	92.3	87.2	41.7	89.6	67.0	90.4	76.5

Table 4: Classification accuracy on different tasks using all neurons (ALL). Re-training: only top/bottom N% of neurons are kept and the classifier is retrained

Linguistic Correlation method are the same as identified by the Cross-model correlation method. Lastly, we found that some of the remaining top Cross-model neurons correspond to fundamental structural properties in a sentence, like relations, conjunctions, determiners and punctuations.

Comparing NMT vs. NLM: There is substantially a large performance difference between top and bottom neurons (Refer to Table 4). For example, averaged over all properties, the top 10% NMT neurons are 12.8% (absolute) better accuracy than the bottom 10% neurons, while the top 10% NLM neurons are 25.5% better than the bottom 10% neurons. We speculate that NMT model distributes the information more, compared to the NLM model. However, this could be an artifact of the difference in the architecture of NLM (unidirectional) and NMT (bidirectional).

Conclusion and Future Work

We proposed two methods to extract salient neurons from a neural model with respect to an extrinsic task or the model itself. We demonstrated the accuracy of our rankings by performing a series of ablation experiments. Our *Cross-model Correlation* method can potentially facilitate research on model distillation and neural architecture search, as it pinpoints what is especially important for the model. Our *Linguistic Correlation* method is primarily focused on trying to understand specific dimensions that are responsible for learning particular properties. This can be helpful for understanding and manipulating systems' behavior. In some preliminary experiments, we were able to successfully manipulate verb tense neurons and control whether the system generates output in present or past tense. Some details are presented in (Bau et al. 2019). The source code for extraction and analysis of salient neurons is incorporated in the NeuroX toolkit (Dalvi et al. 2019) and is available on git.[10]

Acknowledgments

Supplementary Material

Language Property Data: We annotated the date using Tree-Tagger for French POS tags, LoPar for German POS and morphological tags, and MXPOST for English POS tags. For the semantic (SEM) tagging task, we experiment with the lexical semantic task introduced by (Bjerva, Plank, and Bos 2016).[11] We split the available annotated data into 42k sentences for training and 12k sentences for testing.

Results on Morphological Tags: Table 5 shows the results for the classifier performance when masking out neurons for morphological tags. Table 6 shows the results when the classifier is retrained with N% of the neurons.

		Masking-out					
Task	ALL	10%		15%		20%	
		Top	Bot	Top	Bot	Top	Bot
NMT FR (Morph)	88.0	25.2	17.3	39.0	20.3	56.3	24.3
DE (Morph)	87.3	21.8	15.7	33.3	20.8	53.2	29.3
NLM FR (Morph)	90.1	36.3	13.9	45.1	15.5	58.4	19.0
DE (Morph)	86.5	24.2	10.7	40.7	13.0	52.8	19.2

Table 5: Classification accuracy on morphological tags for French and German using all neurons (ALL). Masking-out: all except top/bottom N% of neurons are masked when testing the trained classifier.

		Retraining					
Task	ALL	10%		15%		20%	
		Top	Bot	Top	Bot	Top	Bot
NMT FR (Morph)	88.0	73.5	65.8	78.0	71.6	80.6	75.1
DE (Morph)	87.3	79.3	75.4	82.1	78.9	83.5	80.5
NLM FR (Morph)	90.1	79.5	61.6	82.5	70.3	84.9	75.7
DE (Morph)	86.5	78.3	66.1	81.6	72.4	83.0	77.1

Table 6: Classification accuracy on morphological tags for French and German using all neurons (ALL). Re-training: only top/bottom N% of neurons are kept and the classifier is retrained

References

Adi, Y.; Kermany, E.; Belinkov, Y.; Lavi, O.; and Goldberg, Y. 2016. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *arXiv preprint arXiv:1608.04207*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

---

[10]https://github.com/fdalvi/NeuroX

---

[11]The annotated data is limited to English language only.

Bau, D. A.; Belinkov, Y.; Sajjad, H.; Durrani, N.; Dalvi, F.; and Glass, J. 2019. Identifying and Controlling Important Neurons in Neural Machine Translation. *arXiv preprint arXiv:1811.01157*.

Belinkov, Y.; Durrani, N.; Dalvi, F.; Sajjad, H.; and Glass, J. 2017a. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver: Association for Computational Linguistics.

Belinkov, Y.; Màrquez, L.; Sajjad, H.; Durrani, N.; Dalvi, F.; and Glass, J. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*.

Bjerva, J.; Plank, B.; and Bos, J. 2016. Semantic Tagging with Deep Residual Networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 3531–3541.

Bradbury, J.; Merity, S.; Xiong, C.; and Socher, R. 2017. Quasi-Recurrent Neural Networks. *International Conference on Learning Representations (ICLR 2017)*.

Cettolo, M.; Niehues, J.; Stüker, S.; Bentivogli, L.; and Federico, M. 2014. Report on the 11th IWSLT Evaluation Campaign. *Proceedings of the International Workshop on Spoken Language Translation, Lake Tahoe, US*.

Conneau, A.; Kruszewski, G.; Lample, G.; Barrault, L.; and Baroni, M. 2018. 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 (ACL)*.

Dalvi, F.; Durrani, N.; Sajjad, H.; Belinkov, Y.; and Vogel, S. 2017. Understanding and improving morphological learning in the neural machine translation decoder. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*.

Dalvi, F.; Nortonsmith, A.; Bau, D. A.; Belinkov, Y.; Sajjad, H.; Durrani, N.; and Glass, J. 2019. NeuroX: A toolkit for analyzing individual neurons in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Doshi-Velez, F., and Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. In *arXiv preprint arXiv:1702.08608*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Kádár, Á.; Chrupała, G.; and Alishahi, A. 2016. Representation of linguistic form and function in recurrent neural networks. *arXiv preprint arXiv:1602.08952*.

Karpathy, A.; Johnson, J.; and Fei-Fei, L. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Linzen, T.; Dupoux, E.; and Goldberg, Y. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics* 4:521–535.

Lipton, Z. C. 2016. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*.

Morcos, A. S.; Barrett, D. G.; Rabinowitz, N. C.; and Botvinick, M. 2018. On the importance of single directions for generalization. In *International Conference on Learning Representations*.

Qian, P.; Qiu, X.; and Huang, X. 2016a. Analyzing linguistic knowledge in sequential model of sentence. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Qian, P.; Qiu, X.; and Huang, X. 2016b. Investigating Language Universal and Specific Properties in Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Radford, A.; Jozefowicz, R.; and Sutskever, I. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Shi, X.; Knight, K.; and Yuret, D. 2016. Why Neural Translations are the Right Length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Shi, X.; Padhi, I.; and Knight, K. 2016. Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.

Vylomova, E.; Cohn, T.; He, X.; and Haffari, G. 2016. Word Representation Models for Morphologically Rich Languages in Neural Machine Translation. *arXiv preprint arXiv:1606.04217*.

Wang, Y.-H.; Chung, C.-T.; and Lee, H.-y. 2017. Gate Activation Signal Analysis for Gated Recurrent Neural Networks and Its Correlation with Phoneme Boundaries. *arXiv preprint arXiv:1703.07588*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhou, B.; Khosla, A.; A., L.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. *CVPR*.

Zhou, B.; Sun, Y.; Bau, D.; and Torralba, A. 2018. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*.

Ziemski, M.; Junczys-Dowmunt, M.; and Pouliquen, B. 2016. The United Nations Parallel Corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*.

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* 67:301–320.