TX-Ray: Quantifying and Explaining Model-Knowledge Transfer in (Un-)Supervised NLP

Nils Rethmeier[*] and **Vageesh Kumar Saxena**[*] and **Isabelle Augenstein**[†]

To appear at UAI 2020 – pre-camera-ready version

Abstract

While state-of-the-art NLP explainability (XAI) methods focus on explaining *per-sample* decisions in *supervised* end or probing tasks, this is insufficient to explain and quantify *model knowledge transfer* during (un-)supervised training. Thus, for TX-Ray, we modify the established computer vision explainability principle of 'visualizing preferred inputs of neurons' to make it usable for both *NLP* and for *transfer analysis*. This allows one to analyze, *track and quantify* how *self- or supervised* NLP models first *build* knowledge abstractions in pretraining (1), and then *transfer* abstractions to a new domain (2), or *adapt* them during supervised fine tuning (3) – see Fig. 1. TX-Ray expresses neurons as feature preference distributions to *quantify fine-grained knowledge transfer or adaptation* and *guide human analysis*. We find that TX-Ray can identify prunable neurons for model compression with improved test set generalization and that it can reveal how early stages of self-supervision automatically learn linguistic abstractions like parts-of-speech.

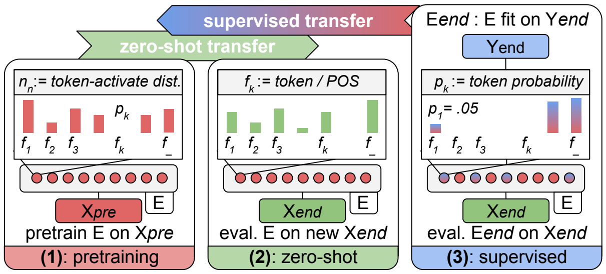

Figure 1: **Example uses of TX-Ray:** for transfer learning and model interpretability. **Left (1):** pre-train a sequence encoder E on corpus X_{pre} and collect feature preference distributions (§2.1, red bars) over input features (e.g. words) f_k.[1] **Middle (2):** apply, but not re-train, the encoder E to new domain inputs X_{end} and observe the *changed neuron activation* (green). Similarities in red and green reveal zero-shot forward transfer potential or data match between X_{pre} and X_{end} according to E. **Right (3):** fine-tune encoder E on supervision labels Y_{end} to reveal 'backward' transfer of supervision knowledge into the encoder's knowledge abstractions.[2]

1 Introduction

Continual and Transfer Learning have gained importance across fields like NLP, where the de facto standard approach is to pretrain a sequence encoder and fine-tune it to a set of supervised end-tasks (Peters et al., 2019). Analysis and understanding of transfer in NLP are currently focused on using either supervised probing tasks (Belinkov and Glass, 2019) to compare task performance metrics (Wang et al., 2019) or laborious per-instance explainability (Belinkov and Glass, 2019). *Supervised* probing annotation is costly, but not guaranteed to be reliable under domain shifts. Probing is also limited to analyzing foreseen (probed) *knowledge absorption* aspects, while unforeseen, model-knowledge properties that underlie and thus further our understanding of self-supervised pretraining remain hidden (McCoy et al., 2019). In fact, 'decision understanding' explainability techniques, as Gehrmann et al. (2019) term them, compute the relevance or impact of a feature or neuron for an end-task prediction score. This makes 'decision understanding' explainability unable to answer the following

[*] DFKI Berlin, Germany, email: first(_first2).last@dfki.de

[†] University of Copenhagen, Denmark, email: augenstein@di.ku.dk

[1] Features f_k are discrete inputs like tokens or POS tags.

[2] 'Backward transfer' since E changes, while labels Y do not.

research questions (**RQ1-3**) – i.e. how can we explain transfer?

(RQ1), unsupervised knowledge absorption: Can explainabilty (XAI) analyze how self-supervised models build and change knowledge abstractions during pretraining and can XAI measure knowledge changes? Do measures coincide with conventional metrics like perplexity? If and when does self-supervision learn linguistic abstractions like word function (parts-of-speech)?

(RQ2), zero-shot knowledge transfer: What knowledge subset do pretrained models apply to a new domain without re-training, e.g. in a zero-shot setting?

(RQ3), supervised/ backwards transfer: Can knowledge transfer 'backwards' from supervision labels into a pretrained model? Does XAI identify which neurons are reconfigured – i.e. become task (ir)relevant due to supervision. Can we validate XAI-based transfer measures (RQ1) empirically by pruning (ir)relevant neurons?

TX-Ray can analyze and *quantify* (self-)supervised model knowledge change: To answer RQ1-3 we propose TX-Ray. TX-Ray – i.e., Transfer eXplainability as pReference of Activations analYsis – modifies the well established activation maximization method of visualizing the preferred inputs of neurons (Erhan et al., 2009) to suit NLP. The resulting fine-grained 'model understanding' – as Gehrmann et al. (2019) term it – enables us to *quantify knowledge changes or transfer* during training at the level of individual neurons – without requiring or preemptively limiting analysis to probing task supervision semantics. The method is designed to explore model knowledge change at both neuron (detail) and model (overview) level to enable concise or deep explorative analysis of unforeseen knowledge transfer mechanics to help us better analyze (continual) transfer, model knowledge generalization (McCoy et al., 2019; Frankle and Carbin, 2019), or low-resource learning. Adebayo et al. (2018); Sixt et al. (2019) showed that XAI methods do not guarantee faithful explanations. We thus use TX-Ray's transfer measures to guide neuron pruning and empirically verify that it can identify task (ir)relevant neurons that boost or lower test set generalization as expected. We also demonstrate that supervision not only causes catastrophic forgetting of knowledge, but also adds new knowledge into previously unpreferred (under-used) neurons (Tab. 2).

2 Approach

TX-Ray is inspired by the widely used activation maximization explainability method, which is based on the idea that "a pattern to which a unit is responding maximally is a good first-order abstraction of what a unit (neuron) is doing. A simple way is to find the input samples that produce the highest activation for a neuron. Unfortunately, this opens the problem of how to 'combine' these samples." (Erhan et al., 2009). In computer vision, naively combining image maximum feature activation maps "over a corpus does not produce interpretable results" (Erhan et al., 2009). In NLP, however, maximal activations of discrete token feature can easily be combined over many samples to form a discrete distribution of 'tokens that a neuron prefers'. These corpus-wide input feature preference distributions let us visualize how each neuron abstracts input knowledge subsets.

A major advantage of using a 'feature preference' method is that it can analyze *non-supervised models over an entire corpus*, while 'prediction score relevance explainability' methods require *supervised models*, and *only explain individual instances* (Belinkov and Glass, 2019). When representing a neuron's abstracted knowledge as a feature preference distributions, we can measure knowledge change or knowledge transfer during learning using standard measures such as Hellinger Distance – i.e., a symmetric version of the Kullback Leibler divergence. This allows one to track changes in neuron knowledge abstractions during model pretraining, model application to new domains or due to supervised fine tuning – see experimental section. Additionally, we automatically determine neurons that change their knowledge the most over time to provide interesting starting points (see Fig. 6, 8) for nuanced, per-neuron analysis (see Fig. 7 and 9).

2.1 Neurons as feature preference distributions:

We thus expresses each neuron n_n as a distribution of features f_k with activation probabilities p_k (Fig. 1) that have been aggregated over an entire corpus to construct each n_n distribution as follows.

(1) Record what features neurons prefer: Given: a corpus D, text sequences $\mathbf{s_i} \in D$, input features (tokens) $f_k \in s_i$, a sequence encoder E, and hidden layer neurons $n_n \in E$, for each input token feature f_k in the corpus sequences s_i, we calculate its: encoder neuron activations $\mathbf{a} = E(f_k)$; along with $\mathbf{a}$'s maximally active neuron $\mathbf{n_{argmax}} = argmax(\mathbf{a})$ and (maximum) activation value $a_{max} = max(\mathbf{a})$; to then record a *single feature's activation* row vector $[f_k, n_{argmax}, a_{max}]$. If the encoder is part of a classifier model C, we also record the sequence's class probability $\hat{y} = C(s_i)$ and true class y as a longer vector $[f_k, n_{argmax}, a_{max}, \hat{y}, y]$. For analyses in RQ1-3, we also record part-of-speech tags (POS, see §3.1) in the row vectors. This produces a matrix M of neuron feature max activations that we aggregate to express each neuron as a probability distribution over maximally activated features in step (2).

(2) Preferred feature distribution per neuron: From rows $m_r \in M$, we generate for each neuron n_n its discrete feature activation distribution $A_{n_n} = \{(f_k, \mu(a_{max_1}, \dots, a_{max_m})) | f_k, n_n, a_{max_j} \in m_r \land m_r \in M \land n_{argmax} = n_n\}$, where each f_k is a feature the neuron maximally activated on, and $\mu(a_{max_1}, \dots, a_{max_m}) = \mu_{f_k}$ is the mean (maximum-)activation of that feature in n_n. We then turn each activation distribution A_{n_n} into a probability distribution P_{n_n} by calculating the sum of its feature activation means $s_{\bar{\mu}} = sum(\mu_{f_1}, \dots, \mu_{f_l})$ and dividing each μ_{f_k} by $s_{\bar{\mu}}$ to produce the normalized distribution $P_{n_n} = \{(f_1, \mu_{f_1}/s_{\bar{\mu}}), \dots, (f_l, \mu_{f_l}/s_{\bar{\mu}})\} = \{(f_1, p_1), \dots, (f_l, p_l)\}\}$, where, each p_{f_k} is now the activation probability of a feature $f_k \in n_n$. Finally, for n neurons in a model, P describes their n per-neuron activation distributions $P = \{P_{n_1}, \dots, P_{n_{n=|E|}}\}$.

Features can be n-grams, and be tracked through multiple layers as in Carter et al. (2019). However, since in this work we focus on concisely presenting TX-Ray's transfer analysis, we only track uni-grams and a single layer.

2.2 Quantify neuron knowledge change as distance:

We use **Hellinger distance** H (Hellinger, 1909) and neuron distribution length l to quantify differences between discrete feature preference probability distributions $p = P_{n_a}$ and $q = P_{n_b}$ of two neurons n_a and n_b as follows:

$$H(p,q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{f_k=1}^{l} (\sqrt{p_{f_k}} - \sqrt{q_{f_k}})^2}; \text{knowl. change}$$

$$l(P_{n_n}) = |\{f_k \mid f_k \in P_{n_n}\}|; \text{knowledge ‘diversity’}$$

Neuron length l describes the number of (unique) maximally activated features in a feature preference distribution P_{n_n}. We use Hellinger distance because it is symmetric, unlike the Kullback-Leibler divergence. Importantly, if one of the preference distributions P_{n_a} or P_{n_b} is empty, i.e. has zero features (zero length), then the resulting Hellinger distance is ill-defined. Thus, Hellinger distance allows one to easily quantify neuron feature preference shifts to measure per-neuron knowledge change during pre-training (RQ1), zero-shot transfer (RQ2), and supervised fine-tuning (RQ3).

Neuron length l on the other hand allows us to define binary states like **‘un-preferred’** for empty preference distributions ($l = 0$) and non-empty ones **‘preferred’** ($l > 0$). We can use the two terms to classify *three kinds of neuron preference state changes caused by different model training stages*: **‘shared’, ‘avoided’, ‘gained’**. For ‘shared’ neurons both distributions are non-empty (preferred) – e.g. when neurons received maximum activations before and after retraining a model. ‘Avoided’ neurons were active ‘preferred’, but became less active ‘un-preferred’ after retraining. Finally ‘gained’ neurons, became more active after retraining, switching from ‘un-preferred’ to ‘preferred’ status. In RQ1-3 we will use changes in Hellinger Distance, distribution length and neuron states to identify which neurons overfit to few preferred features, which ones reuse features (transfer) and which one never specialize (unfit).

3 Experiments and Results

We showcase TX-Ray's usefulness for analyzing and quantifying transfer in answering the previously stated research questions. For RQ1, we pretrain an LSTM sequence encoder E[3] with 1500 hidden units on WikiText-2 similarly to (Merity et al., 2017; Howard and Ruder, 2018), and apply (RQ2) or fine-tune it (RQ3) on IMDB (Maas et al., 2011), so we can analyze its zero-shot and supervised transfer properties. Each RQ's experimental setup and results are detailed below.

3.1 RQ1: How does pretraining absorb knowledge?

In this experiment, we explore how pretraining builds knowledge abstractions. We first analyze neuron abstraction shift between early and later training epochs, and then verify that Hellinger distance and neuron length changes converge similar to measures like training loss.

We pretrain a single layer LSTM encoder E on paragraphs from the WikiText-2 corpus D_{wiki2} using a standard language modeling setup until loss an perplexity converge, resulting in 50 training epochs. We save model states at Epoch 1, 48 and 49 for later analysis. To produce neuron activation distributions P_{wiki_1} (gray), $P_{wiki_{48}}$ (pink) and $P_{wiki_{49}}$ (red) we feed the first 400.000 tokens of WikiText-2 into the Epoch 1, 48 and 49 model snapshots each to compare their neuron adaptation and incremental abstraction building using Hellinger distance and distribution length. Additionally, we record POS feature activation distributions using one POS tag per token, to later group tokens activations by their word function to better read, analyze and compare feature preference distributions – see Fig. 3, 5, 7 or 9. POS tags are produced

[3] Though possible, we do not pretrain Transformers, due to high computation requirements, and since LSTMs encoders perform vastly better when pretraining on standard dataset instead of hugh data collections – compare Wang et al. (2020) with Merity et al. (2017). Instead, we focus on demonstrating TX-Ray's analytical versatility, especially for true low-resource scenarios, where large pre-training is unavailable.

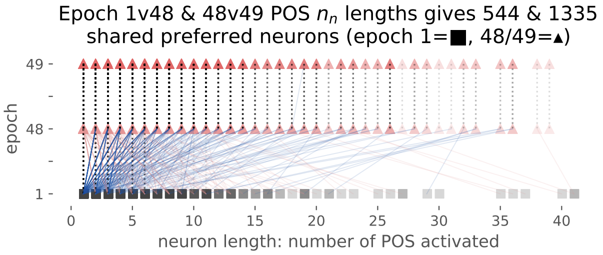

Figure 2: **Pretraining neuron length shifts:** where neuron length l (token variety) becomes; longer (blue /), shorter (red \), unchanged (black :) for epoch 1, 48, 49. Token variety settles (:) in later epochs.

by the state-of-the-art Flair tagger (Akbik et al., 2019) using the Penn Treebank II tag set.

We use this experiment to verify the feasibility of using a feature preference distribution approach, since comparing Epochs 1 vs. 48 should reveal *large changes* to neuron abstractions, while Epoch 48 and 49 should cause *few changes*. The resulting changes in terms of Hellinger distance, amount of 'shared' preferred neurons, and feature preference distribution lengths can be seen in Fig. 2.

While the Epoch 1 vs. 48 comparison produced 544 'shared' neurons, the later 48 vs. 49 comparison shows 1335 'shared' (§2.2) neurons. This means that pretraining the encoder distributes maximum input activations across increasingly many neurons. This can be seen in most neurons becoming longer (blue ■/▲ lines), and fewer neurons becoming shorter (red ▲\■ lines). As expected, for epochs 48 and 49 we see almost unchanged neuron length – seen as dotted vertical (**:**) lines between epochs. Additionally, in later training stages, shorter neurons are more frequent than longer ones, reflected in the opacity of dotted vertical bars decreasing with neuron length. In fact, the average length of 'shared' preferred neurons drops from 944.76 in epoch 1 to 524.55 and 519.34 in epochs 48 and 49.

Since lengths of POS class preference distributions change significantly in the early epochs, we also analyze whether the encoders activations P_{wiki_1}, $P_{wiki_{49}}$ actually learned to represent the original POS tag frequency distribution of WikiText-2. Thus, we express both corpus POS tag frequencies and encoder activation masses as proportional (relative) frequencies per token. In Fig. 3, we see relative corpus POS tag frequencies (black), compared with encoder POS activation percentages for epoch 1 (dark grey) and 49 (red). Evidently, the encoder learns a good approximation of the original distribution (black) even after just the first epoch (dark grey), which confirms findings by Saphra and Lopez (2018), who showed that: "language model pretraining learns POS first", and that "during later epochs (49) the encoder POS representation changes little". Ultimately, the encoder near perfectly replicates the original POS distribution. We thus see that POS are well represented by the encoder, and that neuron adaptation and length shifts converge in later epochs in accordance with the quality of the POS match. This also tells us that TX-Ray, compared with more involved, task-specialized analysis methods (Saphra and Lopez, 2018), can reveal comparably deep insights into the mechanisms of unsupervised training, while being simpler and more versatile (RQ1-3).

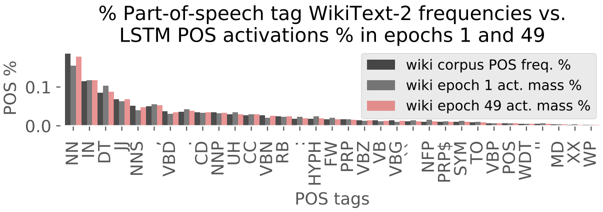

Figure 3: **Encoder learns POS early on:** Black: Corpus tag frequencies (y-axis) vs. encoder activations in %-per-tag. Black: corpus frequencies via FLAIR. Grey: epoch 1 encoder. Red: fully trained encoder. POS is learned early, i.e. in epoch 1, confirming findings in Saphra and Lopez (2018).

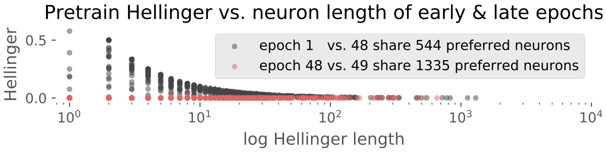

Figure 4: **Encoder gains knowledge preference (neurons) through pretraining:** Later epochs activate more neurons maximally (544 to 1335), while Hellinger distance (knowledge change) reduces in later epochs (48 vs. 49, red line) vs. earlier epochs (1:48, black dots).

Using Fig. 4, a similar analysis about *neuron feature distribution changes stabilizing at later training stages* can be made using Hellinger distances. When visualizing distances, we see that they shrink as expected by 99.92% on average in later epochs and that neuron distance comparisons concentrate on medium length distributions of 10-200 features f_k each. Preference distribution changes of short, specialized, neuron seem to produce higher Hellinger distances than longer, more general neurons. Since distances over different neuron lengths are not and should not be directly compared, this visualization acts to provide an *explorable overview* of neuron distances over different preference distribution lengths, used to identify and examine interesting neurons in *detail*.

[3] https://www.clips.uantwerpen.be/pages/mbsp-tags

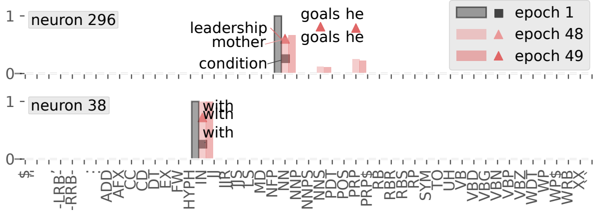

Figure 5: **High vs. low Hellinger H neurons:** Neuron token (■▲▲) and POS activation probabilities (bars) for epochs 1, 48, 49. Neurons with high H (296) and low H (38) between epochs 1:48.

To run such a detail analysis we pick 2 neurons from Fig. 4 for closer inspection of their feature preference distribution changes between Epochs 1, 48 and 49. Fig. 5 thus shows neuron **296** from the top 10 (head) most distant Epoch 1 vs. 48 neurons, and Neuron **38** from the 10 least changed ones (tail). As expected from Neuron **296**'s high Hellinger distances between Epoch 1 and 48, we see that its token and POS distribution for Epoch 1, i.e., an outlined grey bar and the word 'condition' (■), are very different from the Epoch 48 and 49 distributions (▲, ▲), which show no significant change in token and POS distribution – i.e., they look nearly the same. Equally expected from Neuron **38**'s low Hellinger distance for Epoch 1 and 48; we see that it keeps the exact same token, 'with', and POS, 'IN'', across all three epochs. This demonstrates that Hellinger distance identifies neuron change, and that later epochs, as expected, lead to small neuron abstraction changes, while earlier ones, also as expected, experience larger changes.

3.2 RQ2: How does knowledge zero-shot transfer?

In this section, we analyze where and to what extent knowledge is zero-shot transferred when applying a pretrained encoder to text of a new domain – without retraining the encoder to fit that new data.

To do so, we apply the trained encoder E, in prediction-only mode, to both its original corpus IMDB, D_{imdb}, and to the new domain WikiText-2 corpus D_{wiki2}, to generate feature preference distributions P_{imdb} and P_{wiki2} from the encoders' hidden layer, as before. We also record activation distributions for POS, which despite the FLAIR tagger being SOTA across several datasets and tasks, had noticeably low quality on the noisy IMDB corpus. However on the WikiText-2 corpus, tagging produced comparatively sensible results. By comparing neuron token and tag activations P_{imdb} (new domain) vs. P_{wiki2} using Hellinger distances for the same neuron positions as in RQ1, we can now analyze zero-shot transfer as distribution shifts. Put differently, we estimate domain transfer between the pretrained model abstractions and text input from a new domain. High distances between the same neurons in P_{imdb} and P_{wiki2} tell us that the pretrained neuron did not abstract the new domain texts well, resulting in low transfer and poor cross-domain generalization. When comparing P_{imdb} and P_{wiki2} in terms of Hellinger distances vs. neuron lengths in Fig. 6, we see that 1323 out of 1500 pretrained neurons (88.2%) remain 'preferred' ('shared') when applying E to the IMDB domain. A drop in the amount of 'preferred' neurons compared to the RQ1 analysis, though at 1335 to 1323 small, is expected since the pretraining corpus covers a broader set of domains.

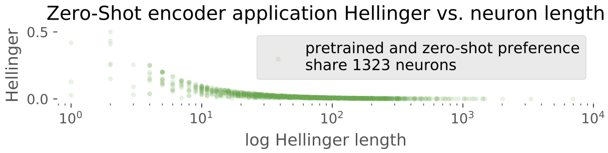

Figure 6: **Neuron feature preference difference when applying to unseen text:** Hellinger distances between neuron **1323 'shared'** preference distributions P_{wiki2} and P_{imdb} on WikiText-2 and IMDB.

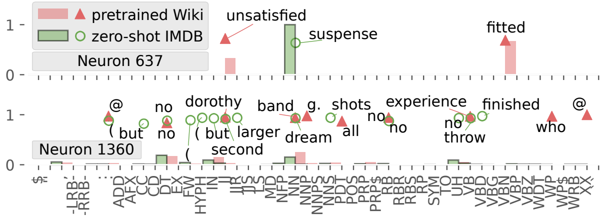

Figure 7: **Low vs. high zero-shot transfer neurons:** Neuron **637** transferred little, while the 'but-no' neuron **1360** transferred (applied) well from pretraining to the new IMDB domain.

However, to gain a *detailed* view of model abstraction behavior and zero-shot transfer, we analyze activation differences between P_{imdb} (green) and P_{wiki2} (red) for two specific neurons, visualizing one each from the 10 most (head) and 10 least (longtail) Hellinger-distant neurons. In Fig. 7 (up), we see Neuron **637**, which has high Hellinger distance when comparing token feature distributions (▲, ○). As expected, the neuron's feature preference between the pretraining corpus P_{wiki2} and the new domain data P_{imdb} changes a lot. In fact, the distance in Neuron 637 is high in terms of both POS classes (word function semantics) and non-synonymous tokens – see x-axis annotated with POS tags and tokens sorted by POS class. Overall, we see very *little knowledge transfer* across data sets within Neuron 637 due to its *feature*

[3] https://plot.ly/python/bar-charts

over-specialization, which is also observable in its short distribution length l – only 2 features activate. When looking at the low Hellinger distance Neuron **1360** in Fig. 7 (lower plot), we see that the neuron focuses on tokens such as 'no' on both datasets and 'but' on IMDB, suggesting that its pretrained sensitivity to disagreement (red), is useful when processing sentiment in the new domain dataset. Furthermore, we see that IMDB specific tokens have many strong activations for movie terms like 'dorothy' or 'shots' (green). We thus conclude that Neuron 1360 is both able to apply (zero-shot transfer) its knowledge to the new domain, as expected from the low Hellinger distance, while also being adaptive to the new domain inputs, despite not being fine-tuned to do so, which is more surprising. In summary, we find that during zero-shot application of an encoder to new domain data, the pretrained encoder exhibits broad transfer, indicated by almost equal amounts of 'shared' neurons between pretraining (1335) and application to the new domain data (1323). A supervision fit encoder however, has its knowledge reconfigured to superivsion, leading to much reduced transfer of pretrained knowledge, as we will describe below in RQ3.

3.3 RQ3: How does supervision (back-)transfer implicit label and text knowledge

In this experiment, we analyze whether transfer constitutes more phenomena than just a high level observation like catastrophic *forgetting*. Here, we want to see if knowledge also transfers 'backwards' from supervised annotations to a pretrained encoder. Specifically, we analyze whether knowledge is added or discarded in two experiments. In Experiment 1, we demonstrate how TX-Ray can identify knowledge addition or loss induced by supervision at individual neuron level (§3.3.1). In Experiment 2, we verify our understanding of neuron specialization and generalization by first pruning neurons that add or lose knowledge during supervision, and then measuring end-task performance changes (§3.3.2). Finally, we show how neuron activity increasingly sparsifies over RQ1-3 to gain overall insights about model-neuron specialization and generalization during unsupervised and supervised transfer (§3.3.3).

For this RQ, we extend the pretrained encoder E with a shallow, binary classifier[4] to classify IMDB reviews as positive or negative while *fine-tuning* E to create a domain-adapted encoder $E_{imdb-sup}$. To guarantee a controlled experiment, we freeze the embedding layer weights and do not use a language modeling objective, such that model re-fitting is exclusively based on supervised feedback – i.e., on knowledge encoded into the labels. We tune the model to produce roughly 80% F_1 on the IMDB test set, to be able to analyze the effects of *even moderate amounts of supervised fine-tuning before task (over-)fitting* occurs. To produce feature preference distributions $P_{imdb-sup}$, we feed the IMDB corpus D_{IMDB} to the newly fine-tuned encoder $E_{imdb-sup}$ – i.e. using the same IMDB text input. We also once more record POS tags for tokens. This time, since POS distributions are compared on the same corpus, their distances are more consistent than in RQ2. Analyzing Hellinger distance and neuron length change when comparing $P_{imdb-sup}$ vs. $P_{imdb-zero-shot}$ will tell us which neuron abstractions were changed the most *due to supervision* – i.e., show us 'backward knowledge transfer'. In Fig. 8, we notice that only 675 neurons were 'shared' compared to 1323 neurons in the zero-shot transfer setting (Fig. 6). In other words, *supervision re-fits the sequence-encoder to 'avoid' (unprefer) nearly half its neurons.*

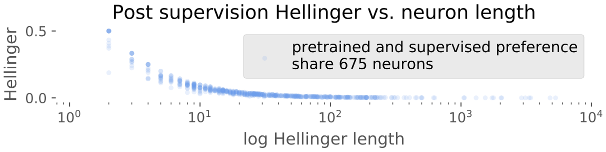

Figure 8: **Neuron feature preference change after supervision:** Hellinger distances of 675 'shared' neuron preferences before and after supervised encoder fine-tuning – dropped from 1323.

3.3.1 Supervision adds and removes knowledge

Somewhat surprisingly, supervision not only erased neurons, but also added distributions for 85 new neurons into $P_{imdb-sup}$ that had previously empty distributions in $P_{imdb-zero-shot}$. We analyzed these neurons and found that they represent new supervision task specific feature f_k detectors. Below in §3.3.1, we show token features f_k for the top three strongest firing neurons n_n and the three least activating neurons out of the 85 – i.e. supervision-specific neurons with the highest or lowest overall activation magnitude. Note: we removed stop-words like 'the' or 'a' as well as spelling duplicates from the table's feature lists to remain brief. Features are sorted by decreasing activation mass from left to right. We see that the first three highly active neurons roughly encode movie-related locations and entities as well as sentiment terms like 'dull' or 'great', though some seem unspecialized (general), fitting many genres.

When looking at the three least activating 'supervision' neurons, we find more specialized feature lists. Some of them are short and very specialized to a specific feature

[4] One fully connected layer with sigmoid activation that is fed by $E's$ end-of-sequence hidden state.

#neuron : activation sum, features, (#features total)
200 : 1307.42 great, james, superb, famous, strange, possible, french, english, grand, final, indian, solid . . . (141)
1210 : 501.97 original, overall, good, real, some, dear, french, british, black, odd, italian, entire, many . . . (161)
125 : 299.12 more, two, best, one, few, most, three, nice, four, fellow, films, somewhat, lot, favorite, rare . . . (77)
1289 : 7.92: terrific, dull, essential, celia, unbelievable, gentle, melancholy, intended, shaggy . . . (14)
372 : 4.18: walter
688 : 0.48: archer

Table 1: **Preferred features of 6/ 85 noisy supervision neurons gained by supervised fine-tuning:** 3 highly active ones (top 3), 3 seldomly active ones (bottom 3).

– e.g. the 372 'walter' neuron seems to be a 'Breaking Bad' review detector, while 'archer' (688) may detect the animated show of the same name. Somewhat surprisingly, Neuron 1289, despite only having a low activation sum, is comprised of many features that focus on sentiment like 'terrific' or 'dull', making the neuron more specialized than the top three. This suggests that 'supervision' neurons with low activation mass, somewhat independent of their feature variety, are more specialized than the highly active ones – which reflects in their lower 'neuron length', i.e. them preferring fewer features.

Additionally, as done by explainability methods, we can approximate how important input features are for correct classification by (re-)weighting, i.e. multiplying each feature f_k in an input sample, with its class prediction probability. When doing so (not shown), the 85 neuron's features reorder to disfavor review score irrelevant terms like numeric expressions (neuron 125). Detailed 'discoveries' like supervision-gained knowledge reinforce our motivation, that an exploration-investigation approach can reveal detailed insights about a model's inner workings if 'drilled-down'[5] far enough, which underlines TX-Ray's application potential.

3.3.2 Pruning avoided, shared and gained neurons

To understand how much the 'avoided', 'shared' and 85 neurons 'gained' by supervision affect predictive task performance, we run four pruning experiments (A-D) that remove neuron sets to measure the relative change from the unpruned F_1 score in % – i.e., a drop from 80 to 77 is $^{77-80}/_{80} = -3.75\%$. Experiment (A) cuts 740 'avoided' neurons from the encoder $E_{imdb-sup}$, i.e., 740 neurons with empty feature preference distribution after supervision. Experiments B and C cut the 20 least and most active neurons from the supervision tuned encoder. To select 20 neurons each, we sort neurons by their individual activation mass, i.e. the sum of a neuron's (max) activations, where 'unpreferred' neurons with an empty preference distribution have zero activity. In the last pruning experiment (D), we prune the 85 neurons that became 'preferred' after (due to) supervision – i.e., were 'unpreferred' before in $P_{imdb-zero-shot}$. Tab. 2 shows for each pruning: the relative changes in training and test set F_1 and what percentage of the encoders entire (max) activation mass the pruned neurons drop.

Which neurons prunded?	**% AM of 675**	**F1 change %**		**pruning effect**
		train	**test**	
none = baseline	100.000	0.00	0.00	–
A: 740 **avoided**	–	3.65	2.80	↓ noise, ↑ generality
B: 20 least **prefered**	0.004	-3.79	0.00	↓ over-fitting
C: 20 top **prefered**	83.120	-4.99	-1.43	↓ generalization
D: 85 **sup gained**	3.006	-3.71	-3.87	↓ sup. knowledge

Table 2: **Pruning avoided, preferred and supervision-gained neurons:** After supervised encoder fitting; (A) prunes avoided (unpreferred) neurons, (B,C) prune the least and most preferred neurons, and (D) prunes 85 neurons gained by supervision – i.e., that were non-preferred in pretraining. Colors represent relative score change in % from original – score drops (red,$-$), gains (blue).

For pruning experiment **(A)**, we see that removing 'avoided' neurons not only does not drop performance as commonly observed when dropping irrelevant neurons (Voita et al., 2019), but actually *increases both training and test set* performance by 3.65 and 2.80 % respectively, resulting in better generalization – at least as far as test set scores reflect generalization. In Experiment **(B)**, when removing seldomly activated supervision neurons, as indicated by the low activation mass percentage of 0.004%, we lose significant training performance (-3.79%), but no test set performance, telling us that those neurons were over-specialized or over-fit to the training set. It also tells us that these neurons were likely short (over-specialized), similar to those in §3.3.1 that have low activation mass (372, 688). When we examined this intuition, we found that each of the 20 neurons has a length of exactly one – i.e. is over-specialized. When pruning the 20 most heavily used supervision neurons **(C)** with 83.12% (max) activation mass, we see the largest drop in training set performance out of all experiments (A-D). This tells us that, similar to observations in experiment (B), TX-Ray again identified neurons that strongly over-fit to the training data, while they overfit the test set to a lesser extend. Thus, Experiments (B, C) indicate that cutting supervision specific

[5] A fundamental visualization techniques design pattern used to describe incrementally more focused analysis.

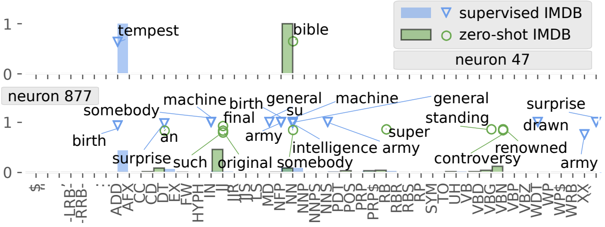

Figure 9: **Low and transfer to supervision:** Neuron **47** saw no transfer, while Neuron **877** transferred its knowledge better from before (zero-shot) to after supervised fine-tuning on IMDB.

neurons after training can help preserve generalization performance, i.e., reduce generalization loss. Lastly, for **(D)**, when pruning the 85 neurons 'gained' by supervision both training and test performances drop by equal amounts. Since these 85 supervision-only neurons only became 'preferred' after supervised fine-tuning, this indicates that pretraining-exposed neurons as in (B) and (C), suffer less from overfitting on new (test set) data, even when pruned. We reason that pretraining-exposed neurons in (B) and (C) have their knowledge partially duplicated across other neurons, while the supervision-only knowledge in the 85 'gained' neurons (D) has no such backups. **(Neuron) generalization, specialization:** These observations are not only consistent with known effects of pretraining on generalization (Peters et al., 2019; Howard and Ruder, 2018), but also show that TX-Ray can identify and distinguish at *individual neuron level*, which parts of a neural network improve or preserve generalization (A, B) and which do not (C, D). Moreover; though the results in Experiment (A, B) initially contradict established views on pruning (Voita et al., 2019), i.e. that it should lead to a slight performance drop, they are perfectly consistent with the notions of *neuron specialization an generalization* used throughout the analysis with TX-Ray. This demonstrates the method's effectiveness in identifying neurons that affect generalization and specialization.

To again analyze what individual neurons learned, we inspect neurons with high and low Hellinger distances between encoder activations before (green) $P_{imdb-zero-shot}$ and after supervision (blue) $P_{imdb-sup}$. In Fig. 9, we show Neuron **47** (up), from the top 10 highest Hellinger distances. We see that the neuron 47 changed in both POS and token distributions after supervision, which suggests catastrophic forgetting, or supervised reconfiguration. For the low Hellinger distance Neuron **877** (down), we see some POS and token distribution overlap before and after supervision, and that movie review related terms (green ▽) become relevant, compared to noticeably war related tokens before supervision (green ○). This shows the neuron's semantic shift (POS, token) due to supervision – i.e., limited knowledge transfer occurred despite the low Hellinger distance. Moreover, distribution length changed for this neuron from 9 before to 15 tokens after supervision, indicating a lack of transfer. Finally, we recall that in the zero-shot case more neurons were 'shared' than after supervision, 1323 vs. 675 (Fig. 6 vs. Fig. 8), which should be reflected in the overall activation magnitude produced by encoder E before and after supervision.

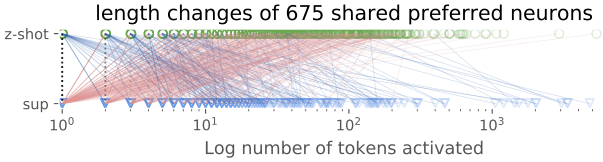

Figure 10: Neuron length before (○) and after **sup**ervision (▽). After supervision / neurons are shorter, \ are longer, and : are unchanged.

3.3.3 Supervision sparsifies neuron knowledge

To investigate the distribution length shift and activation sum hypotheses formulated above, we visualize the shift of neuron length before and after supervision (Fig. 10 and Fig. 11), as well as the activation mass for the three research questions: (RQ1) pretraining, (RQ2) zero-shot, and (RQ3) supervision.

In Fig. 10, we see neurons that shortened (red lines, ▽/○), or got longer (blue lines, ○\▽), after supervision. Token preference distributions of neurons actually *slightly lengthen* by 4.62% on average over the 675 shared neurons,[6] while POS preference distributions, severely shorten at 32.83% (not shown). Similar neuron lengthening, 'feature variety increase', from supervision, was already apparent in neuron 877 (Fig. 9), where supervision appeared to have specialized and extended a previously unspecific neuron into a movie sentiment detector[7].

In Fig. 11, we see that the activation mass – i.e., the sum of activation values – differs across corpora and encoder activation distributions $P_{imdb-zero-shot}$, $P_{imdb-sup}$ and P_{wiki2}. A much more peaked activation mass is produced after the encoder has been fine-tuned via supervision and then again applied to IMDB (blue, ▽) compared to before supervision (green), which is a strong in-

[6] Over the entire 1500 neurons, neuron token length shortens by 42.53% after supervision.

[7] Again, without deeper analysis, we are not claiming that this is the case, only that such points for investigation and new, interesting hypotheses can be identified via TX-Ray.

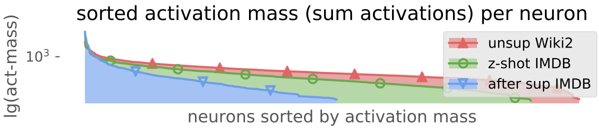

Figure 11: Sorted neuron activation masses, for the pretrained (large, ▲), zero-shot (middle, ⊖), and supervision tuned encoder (small, ▽). Supervision sparsifies activations – i.e. ▽ head peaks, tail shortens.

dicator that supervision sparsified the neuron activation and therefore the abstractions in the encoder. The activation mass of the pretrained encoder E on its pretraining corpus (WikiText-2, red ▲) is, unsurprisingly, the broadest, while the same encoder E activates less strongly on the same amount of text (400k tokens) on the IMDB text (green, ⊖), due to the mismatch of domains between pretrained encoder and the new data domain – as previously detailed in RQ2.

4 Related Work

From summarizing recent explainability methods (Gehrmann et al., 2019; Belinkov and Glass, 2019; Gilpin et al., 2018), two kinds of approaches emerge: *supervised* 'model-understanding (MU)' and 'decision-understanding (DU)'. DU treats models as black boxes by visualizing how important each input is for a prediction outcome to understand model decisions. MU, enables a grey-box view by visualizing internal model abstractions to understand what knowledge a model learned. Both DU and MU heavily focus on analyzing supervised models, while understanding transfer learning in self- and supervised models remain open challenges. **Supervised 'DU':** techniques use probing tasks to *hypothesis test* models for language properties like syntax and semantics (Conneau and Kiela, 2018), or language understanding (Wang et al., 2019; Giulianelli et al., 2018). DU is limited to *supervised* analysis of *individual* samples (Gilpin et al., 2018; Arras et al., 2019). **MU:** techniques like Activation Atlas or Summit (Carter et al., 2019; Hohman et al., 2020) explore supervised model knowledge in vision, while NLP methods like Seq2Seq-Vis (Strobelt et al., 2019) compare model behavior using many per-instance explanations. However, these methods produce a high cognitive load, showing many details, which makes it harder to understand overarching learning phenomena. **(Un-) supervised 'model and transfer understanding':** TX-Ray modifies ideas behind activation maximization (Erhan et al., 2009; Olah et al., 2017; Carter et al., 2019) (see §2) to enable measuring neuron knowledge change, specialization and generalization as well as to guide explorative transfer analysis by *quantifying interesting starting points*. Somewhat similarly to our setup in RQ3, Singh et al. (2019) "calculate Helliger distances over 'neuron feature dictionaries‘ to measure neuron adaptation during 'supervised' task learning" in the prefrontal cortex of rats. Measuring changes in neuron feature preference distributionsenables fine-grained analysis of neuron (de-)specialization and model knowledge transfer in RQ1-3. TX-Ray thus presents a novel (un-)supervised transfer interpretability method (Belinkov and Glass, 2019; Gilpin et al., 2018), that supports deep analysis of transfer in current and future (continual) pretraining methods (Peters et al., 2019; de Masson d'Autume et al., 2019) as well as discovery of unforeseen hypotheses to help scale learning analysis beyond probing task limitations.

5 Conclusion and Future Work

We presented TX-Ray, a simple, yet nuanced model knowledge explainability method for analyzing how neuron knowledge transfers between pretraining (RQ1), zero-shot knowledge application (RQ2), and supervised fine-tuning (RQ3). We showed how to extract neuron knowledge abstractions in NLP, developed extensible explainability visualizations and demonstrated how this can measure knowledge abstraction change. We find that TX-Ray enables explorative analysis of how knowledge is lost and added during supervision (RQ3), how neurons overfit or generalize (RQ1-3), and how pretraining builds knowledge abstractions (RQ1). TX-Ray is designed to reduce computational and cognitive load, but is flexible and scalable to future visualizations and analysis. In future, we will extend and explore using TX-Ray in more advanced transfer tasks, model settings and with more activations and metrics. Code for TX-Ray and the presented visualization types is available in Weights & Biases via `anonymized_for_review`.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I. J., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. In *Advances in the 31: NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of NAACL 2019 (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. ACL.

Arras, L., Osman, A., Müller, K.-R., and Samek, W. (2019). Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP*, pages 113–126, Florence, Italy. ACL.

Belinkov, Y. and Glass, J. (2019). Analysis methods in neural language processing: A survey. *Transactions of the ACL*, 7:49–72.

Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. (2019). Activation atlas. *Distill*.

Conneau, A. and Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of LREC 2018, Miyazaki, Japan, May 7-12, 2018*.

de Masson d'Autume, C., Ruder, S., Kong, L., and Yogatama, D. (2019). Episodic memory in lifelong language learning. In *Advances of NeurIPS 2019, 8-14 December 2019, Montréal, Canada*.

Erhan, D., Bengio, Y., Courville, A. C., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. In *University of Montreal publications*.

Frankle, J. and Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Gehrmann, S., Strobelt, H., Krüger, R., Pfister, H., and Rush, A. M. (2019). Visual interaction with deep learning models through collaborative semantic inference. *IEEE TVCG*, pages 1–1.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *5th IEEE DSAA 2018, Turin, Italy, October 1-3, 2018*, pages 80–89.

Giulianelli, M., Harding, J., Mohnert, F., Hupkes, D., and Zuidema, W. H. (2018). Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the BlackboxNLP@EMNLP 2018 Workshop, Brussels, Belgium, November 1, 2018*, pages 240–248.

Hellinger, E. (1909). Neue Begründung der Theorie Quadratischer Formen von Unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271.

Hohman, F., Park, H., Robinson, C., and Chau, D. H. (2020). Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE TVCG*.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th ACL-HLT*, pages 142–150, Portland, Oregon, USA. ACL-HLT.

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3428–3448.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *5th ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization. *Distill*. https://distill.pub/2017/feature-visualization.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th RepL4NLP@ACL 2019 Workshop, Florence, Italy, August 2, 2019.*, pages 7–14.

Saphra, N. and Lopez, A. (2018). Language models learn POS first. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 328–330.

Singh, A., Peyrache, A., and Humphries, M. D. (2019). Medial prefrontal cortex population activity is plastic irrespective of learning. *Journal of Neuroscience*, 39(18):3470–3483.

Sixt, L., Granz, M., and Landgraf, T. (2019). When explanations lie: Why modified BP attribution fails.

Strobelt, H., Gehrmann, S., Behrisch, M., Perer, A., Pfister, H., and Rush, A. M. (2019). Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE TVCG*, 25(1):353–363.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th ACL*, pages 5797–5808, Florence, Italy. ACL.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Wang, C., Ye, Z., Zhang, A., Zhang, Z., and Smola, A. J. (2020). Transformer on a diet.