A SOLUTION FOR LARGE-SCALE ANALYSIS ON TWEETS: LEVERAGING INTERMEDIATE NEURON ACTIVATIONS FROM PRETRAINED LARGE LANGUAGE MODELS

Yilun Liu
Technical University of Munich
yilun.liu@tum.de

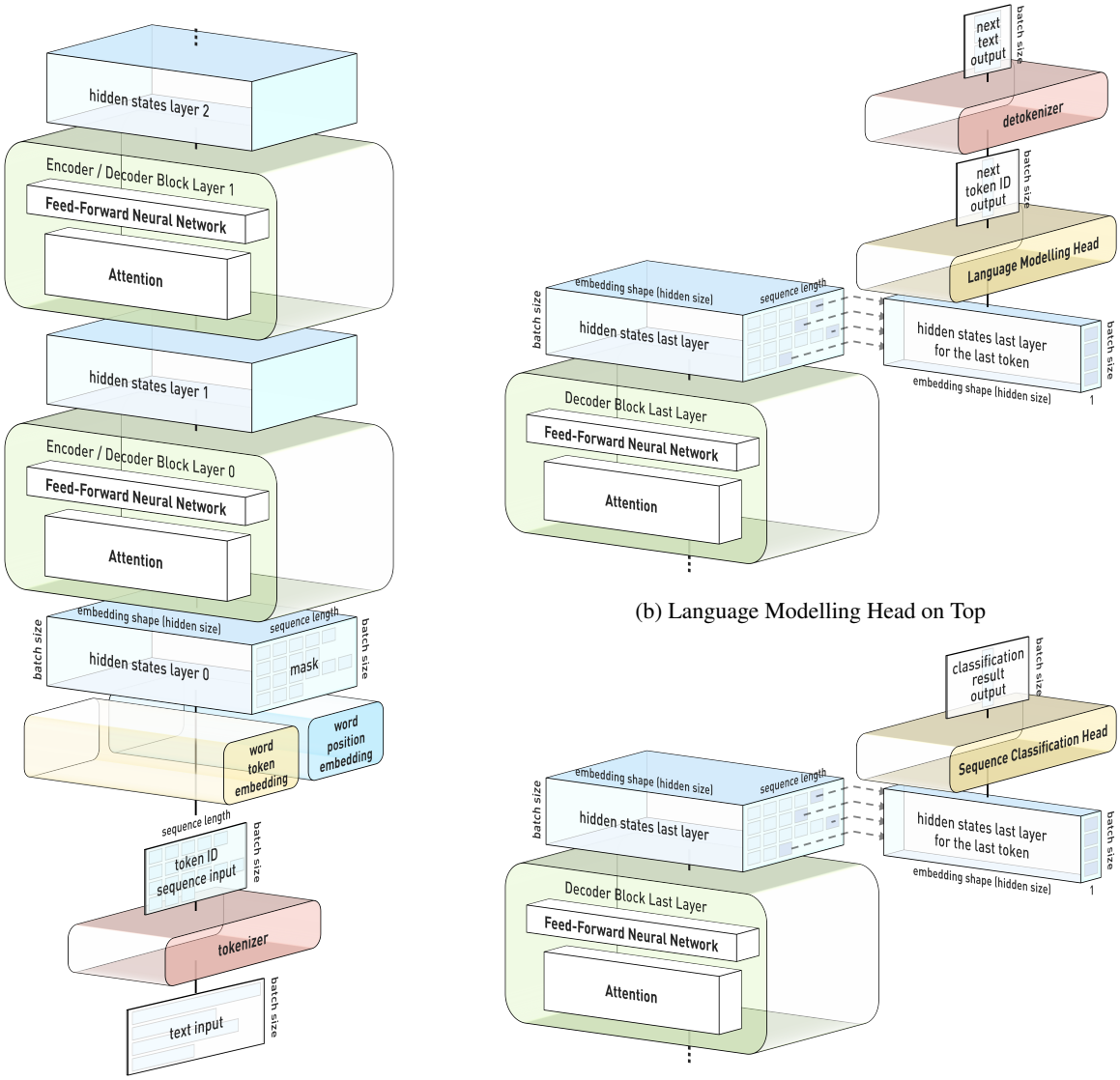
ABSTRACT

Recent advancements in Large Language Models (LLMs) have led to significant improvements in a myriad of complex Natural Language Processing tasks. While both dominant transformer-based approaches in LLMs, the BERT and GPT families, share architectural similarities, their specialized expertise in task categories confines their downstream application scenarios, especially for large-scale analysis on our Twitter datasets. This application project introduces a method utilizing the activation patterns of feed-forward neuron networks within the transformer blocks at different layers of pretrained LLMs to execute text classification and regression tasks directly. Contrasting the conventional methodology that requires fine-tuning task-specific heads using the final layer of hidden states, our approach focuses on selection of salient neurons and a lightweight knowledge-consolidation head designed atop their activations. In applications, our refined technique, by only requiring running the model to specific levels, could not only conserve a significant amount of computational resources but also ensure purer outputs are minimally affected by irrelevant components within the original models. Preliminary experimental results on models including RoBERTa, DistilBERT, and GPT2-XL demonstrate that our approach could already surpass the performance of traditional classification heads by only leveraging their neuron activation results from lower layers.

1 INTRODUCTION

Large Language Models (LLMs), though being simply trained on textual data to predict the masked or next tokens, have in recent years become remarkably advanced tools with emergent capabilities that come close to or even exceed human performances among a wide range of rather complex Natural Language Processing (NLP) tasks, including natural language inference, machine translation, text summarization, speech recognition, and question answering. The current mainstream transformer-based models can predominantly be categorized into two primary approaches: BERT-like models (RoBERTa, DeBERTa, DistilBERT, etc.), and GPT-like models (GPT2, GPT3, GPT4, etc.). Respectively, these models represent two distinct methodologies in terms of their architectures and training objectives: BERT for bidirectional language understanding, and GPT for auto-regressive language modeling and generation, i.e. to recursively predicting the next word or token in a sequence of preceding context.

A foundational structure for both families of LLMs comprises three principal components: an embedding layer focusing on token representation into dense vectors within a high-dimensional hidden space, multiple encoder blocks (for BERT family) or decoder blocks (for GPT family) with attentions and feedforward neural networks that jointly apply nonlinear transformations on hidden representations, and a final head layer that that reshapes and refines the transformed representations to produce the ultimate output.



(a) Embedding Layer and Decoder Blocks

(b) Language Modelling Head on Top

(c) Sequence Classification Head on Top

Figure 1: General Architecture of Large Language Models in GPT Family

Though having explicit architectures and application paradigms, the implicit processes by which these models achieve their remarkable capabilities, stemming from the vast amount of weights, remain subjects of research. A pivotal challenge is to understand the way they embed and process the syntactic, semantic, and pragmatic information of training and test data into the model. One way to probe such representations learned by these models involves analyzing the activation patterns of the neurons within the Feed-Forward Neuron Networks (FFNs) of each encoder or decoder block. Such patterns are often seen as responsible for capturing the input text's intermediate representations at different levels. Intuitively and empirically, by scrutinizing these activations, we can gain valuable insights into the specific linguistic features and structures of the textual input internalized by the language models, and how these may differ across models and contexts, which could be utilized respectively for application in downstream tasks.

In our experiment, we start by leveraging the activations of the FFN neurons at different layers of the pretrained GPT2-XL model to implement task-specific classifiers. From the massive activations hidden within block

layers, we identify salient neurons relevant to specific tasks and subsequently develop classifiers emphasizing the activations of salient neurons. Current preliminary results suggest that our proposed method notably outperforms the original paradigm of GPT2-XL sequence classification.

This work make three main contributions:

1. We verify that for both transformer based LLMs BERT and GPT families, the neurons within different layers hold robust capabilities in capturing vital syntactic, semantic, and pragmatic information of the textual input respectively.
2. We propose a lightweight method that leverages activation patterns of identified **salient neurons** from feed-forward neuron networks within transformer blocks across different layers of pretrained LLMs to form **knowledge-consolidation heads** for text classification tasks. .
3. We show that our proposed method, even when only considering neuron activations from lower layers, can outperform traditional classification heads. Tests were carried out on well-established models such as RoBERTa, DistilBERT, and GPT2-XL.

2 METHODOLOGY

2.1 DATA PREPARATION

The canonical structure of Transformer-based large language model consists of L identical encoder or decoder block layers, each of which contains a K-head (self-)attention mechanism and a feed-forward network. The intermediate representations in-between blocks are of H-dimensional hidden space.

Raw texts tailored for specific tasks serve as inputs to the pretrained tokenizers before entering the language model. After going through model-specific word-token embedding and word-position embedding mechanisms, a tokenized input sequence of length N becomes the first hidden state $\boldsymbol{X}_0 \in \mathbb{R}^{N\times H}$ that works as the input to the block at layer 0. Afterwards, each transformer block at layer l can be formulated as follows:

$$\boldsymbol{Q}_{kl} = \boldsymbol{X}_l\boldsymbol{W}_{kl}^Q, \boldsymbol{K}_{kl} = \boldsymbol{X}_l\boldsymbol{W}_{kl}^K, \boldsymbol{V}_{kl} = \boldsymbol{X}_l\boldsymbol{W}_{kl}^V \tag{1}$$

$$\text{Attn}_{kl}(\boldsymbol{X}_l) = \text{softmax}(\boldsymbol{Q}_{kl}\boldsymbol{K}_{kl}^T)\boldsymbol{V}_{kl} \tag{2}$$

$$\text{FFN}_l(\boldsymbol{H}_l) = f_{\text{act}}(\boldsymbol{H}_l\boldsymbol{W}_{1l}^T)\boldsymbol{W}_{2l} \tag{3}$$

where matrices with $k \in \{0, \cdots, K\}$ represents different attention heads, $\boldsymbol{H}_l$ being formed by projecting the attention output concatenation of all heads. The scaling factors, layer norms, dropout and bias terms are omitted here for simplicity.

Activation Acquisition In particular, we focus on the activation results within $\text{FFN}_l(H_l)$, represented as

$$\boldsymbol{A}_l = f_{\text{act}}(\boldsymbol{H}_l\boldsymbol{W}_{1l^T}) \in \mathbb{R}^{N\times A} \tag{4}$$

in which $A = kH$ represents the number of neurons, i.e., the dimension of hidden space, within the FFN, and in practice for most models $k = 4$. Given certain activation records among all layers $\boldsymbol{A} = \{\boldsymbol{A}_0, \cdots, \boldsymbol{A}_{L-1}\} \in \mathbb{R}^{L\times N\times(kH)}$, it is subsequently condensed token-wise by the average and max pooling function

$$g(\boldsymbol{A}_l, f_{\text{pooling}}) : \mathbb{R}^{N\times A} \to \mathbb{R}^{2\times A} \tag{5}$$

to express the aggregated activation of each neuron across the full input sequence. The distilled activations are stored for the following processing.

In practice, the model is most frequently running on hidden states in batches $\boldsymbol{X}_l \in \mathbb{R}^{B\times N\times H}$ along with masks for attention $\boldsymbol{M} \in \{0, 1\}^{B\times N}$, in which B is the batch size and N length of the longest token sequence within the batch. In this case, the pooling function should also incorporate the mask to calculate the correct average and max value within the range of sentences.

We have already successfully replicated the entire forward pass of RoBERTa[1], DistilBERT[2], and GPT2-XL[3] model to access their intermediates and activations, and integrating them with pretrained weights sourced

[1]For RoBERTa, $A = 768 \times 4, H = 768, K = 12, L = 12, \max(N) = 512, f_{\text{act}} = \text{GELU}$.
[2]For DistilBERT, $A = 768 \times 4, H = 768, K = 12, L = 6, \max(N) = 512, f_{\text{act}} = \text{GELU}$.
[3]For GPT2-XL, $A = 6400, H = 1280, K = 16, L = 48, \max(N) = 1024, f_{\text{act}} = \text{GELU}_{\text{new}}$.

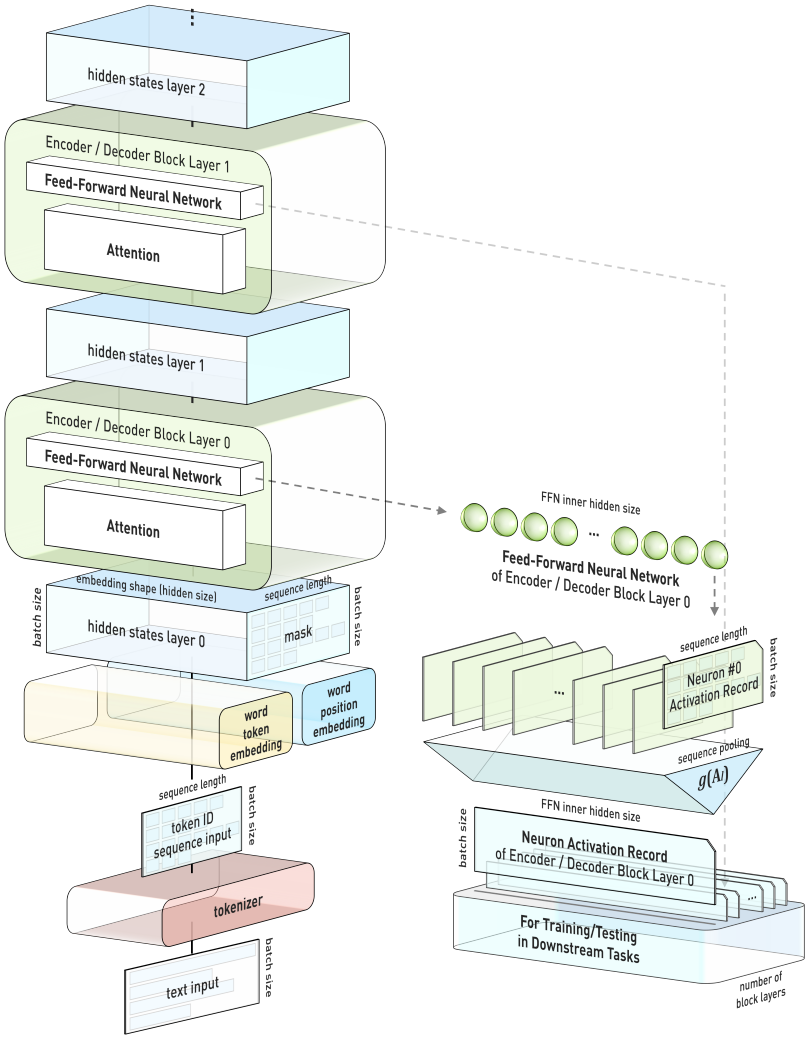


Figure 2: Proposed Architecture for Neuron Activation Records Acquisition

from HuggingFace. Within each encoder or decoder block in the models, the activations of the Feed-Forward Neural Network, which comprises A neurons in each block, are collected, pooled, and stored for the following processing.

Salient Neuron Identification Here we proposed a dual-phase training architecture for a task-specific head, progressively serving as salient neuron indicators, as feature extractors from neuron activation results, and as the desired task resolver at the end across its two phases. During Phase 1, the head is designed as a series of independent, single-layer fully-connected neuron networks. Each operates on activation records from different LLM block layers and all are trained for the same target task. This mirrors a fine-tuning process on the subsequent layers ahead of the studied layers, without actually modifying any original model parameters, as to retain the LLM's core capabilities to the fullest extent and at the same time being more efficient in data and computation required. More importantly, after being properly trained, these single-layer networks would act as indicator of salient neurons, by investigating the overall importance (weights quantified by the $\mathcal{L}_2$-norm) of connections from different FFN neurons' input records of each block layer, as a higher overall weight for a FFN neuron signifies its substantial contribution within the trained selector to the target task's results.

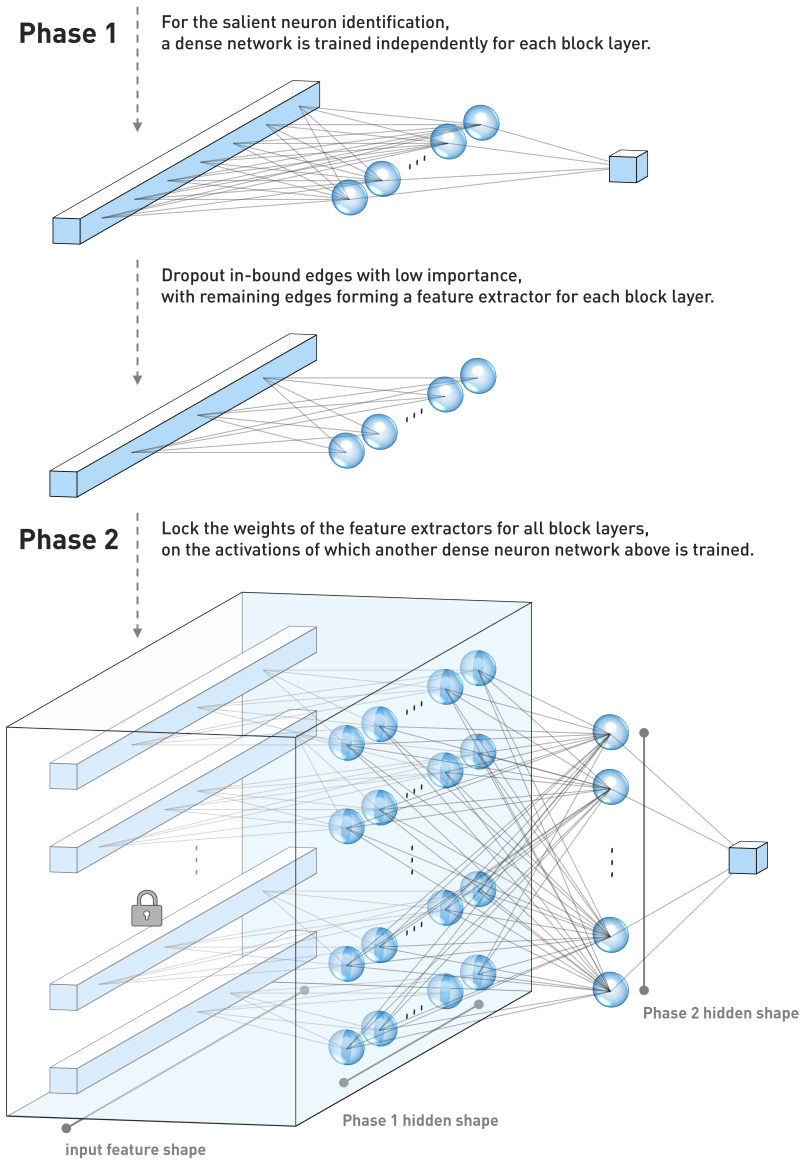


Figure 3: Proposed Dual-Phase Architecture of the Task-Specific Head for Sequence Classification Using FFN Activation Records from LLM Blocks

Knowledge Consolidation In Phase 2, we leverage the identified salient neurons along with the trained weights acquired during Phase 1 to train a refined task resolver network at a higher level. This incorporates a guided dropout of input edges with low importance for each layer, freezing the weights of remaining input edges, and training the cross-layer structure above for the target task. A dense layer fully-connects all the neurons from the previous single-layer networks to the secondary hierarchy of neurons that integrate information across different LLM block layers. At this time, after dropping out the edges from features of unimportant FFN neurons, the frozen pretrained sub-networks for each layers in Phase 1 would act as task-specific feature extractors in turn, providing a higher level of activation records on each GPT block layers for the cross-layer network above to aggregate.

When training for the desired task in Phase 2, we could consider using both previously seen and unseen data in Phase 1. For new data with the same distribution as the seen data, the feature extractors should work well as

they were trained on the same tasks and have already embody the necessary knowledge, and the networks above would benefits from this consistent intermediate results. For the used data on which the extractors were already trained, there would be no over-fitting problems for feature extractors as their weights are frozen, and training the network above with these features should further consolidate the patterns that the extractors have learned.

3 Experimental Result on IMDb Sentiment Classification Dataset

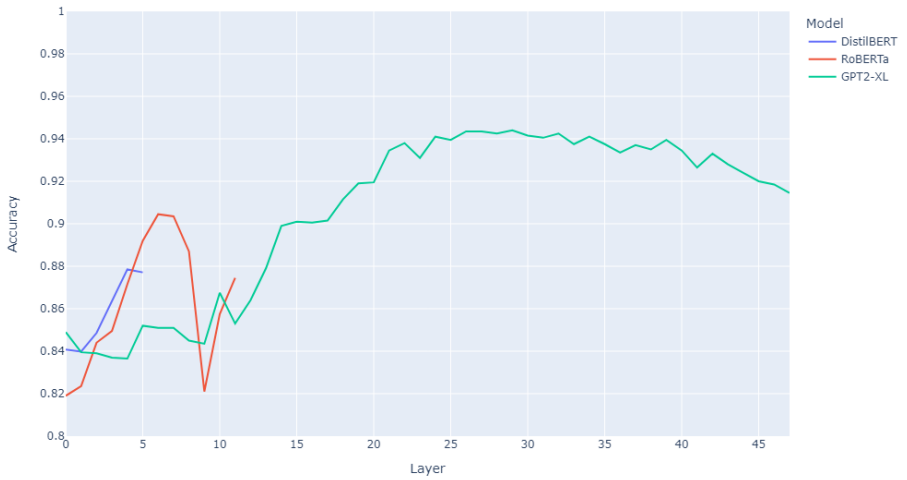


Figure 4: IMDb sentiment classification accuracy result, with linear regression on single-layer activation from different models

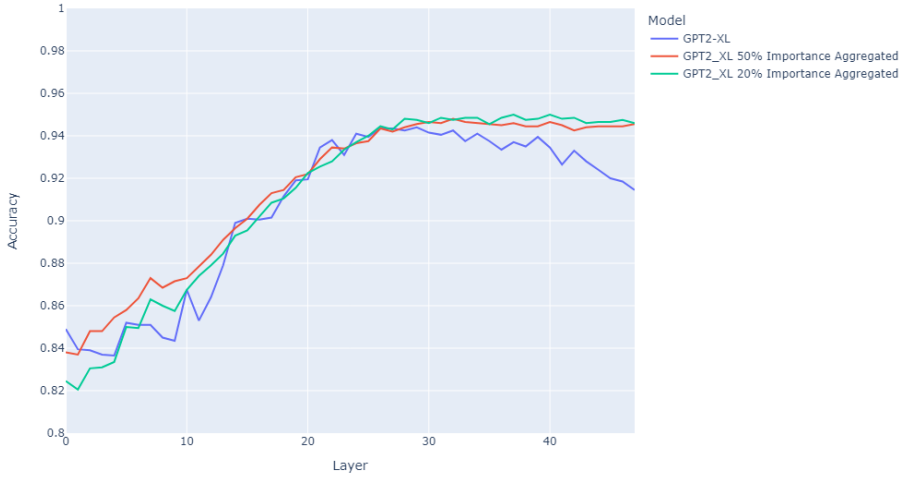


Figure 5: IMDb sentiment classification accuracy result, with linear regression on aggregated GPT2-XL layer activation based on single-layer results

REFERENCES

S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2023.

C. Burns, H. Ye, D. Klein, and J. Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

S. Casper, Y. Li, J. Li, T. Bu, K. Zhang, K. Hariharan, and D. Hadfield-Menell. Red teaming deep neural networks with feature synthesis tools, 2023.

D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

W. Gurnee and M. Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

W. Gurnee, N. Nanda, M. Pauly, K. Harvey, D. Troitskii, and D. Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

H. R. Kirk, W. Yin, B. Vidgen, and P. Röttger. Semeval-2023 task 10: Explainable detection of online sexism. *arXiv preprint arXiv:2303.04222*, 2023.

M. Li, X. Davies, and M. Nadeau. Circuit breaking: Removing model behaviors with targeted ablation, 2023.

A. Panigrahi, N. Saunshi, H. Zhao, and S. Arora. Task-specific skill localization in fine-tuned language models. *arXiv preprint arXiv:2302.06600*, 2023.

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

T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 464–483. IEEE, 2023.

N. Stoehr, P. Cheng, J. Wang, D. Preotiuc-Pietro, and R. Bhowmik. Unsupervised contrast-consistent ranking with language models. *arXiv preprint arXiv:2309.06991*, 2023.

A. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

X. Wang, K. Wen, Z. Zhang, L. Hou, Z. Liu, and J. Li. Finding skill neurons in pre-trained transformer-based language models. *arXiv preprint arXiv:2211.07349*, 2022.

H. Ye, J. Zou, and L. Zhang. Freeze then train: Towards provable representation learning under spurious correlations and feature noise. In *International Conference on Artificial Intelligence and Statistics*, pages 8968–8990. PMLR, 2023.