Explainability for Large Language Models: A Survey

Haiyan Zhao[1] *hz54@njit.edu*
Hanjie Chen[2,8] *hanjie@rice.edu*
Fan Yang[3] *yangfan@wfu.edu*
Ninghao Liu[4] *ninghao.liu@uga.edu*
Huiqi Deng[5] *denghq7@sjtu.edu.cn*
Hengyi Cai[6] *hengyi1995@gmail.com*
Shuaiqiang Wang [7] *wangshuaiqiang@baidu.com*
Dawei Yin [7] *yindawei@acm.org*
Mengnan Du[1] *mengnan.du@njit.edu*

[1]*New Jersey Institute of Technology* [2]*Johns Hopkins University* [3]*Wake Forest University* [4]*University of Georgia*
[5]*Shanghai Jiao Tong University* [6]*Institute of Computing Technology, CAS* [7]*Baidu Inc.* [8]*Rice University*

Abstract

Large language models (LLMs) have demonstrated impressive capabilities in natural language processing. However, their internal mechanisms are still unclear and this lack of transparency poses unwanted risks for downstream applications. Therefore, understanding and explaining these models is crucial for elucidating their behaviors, limitations, and social impacts. In this paper, we introduce a taxonomy of explainability techniques and provide a structured overview of methods for explaining Transformer-based language models. We categorize techniques based on the training paradigms of LLMs: traditional fine-tuning-based paradigm and prompting-based paradigm. For each paradigm, we summarize the goals and dominant approaches for generating local explanations of individual predictions and global explanations of overall model knowledge. We also discuss metrics for evaluating generated explanations, and discuss how explanations can be leveraged to debug models and improve performance. Lastly, we examine key challenges and emerging opportunities for explanation techniques in the era of LLMs in comparison to conventional deep learning models.

1 Introduction

Large language models (LLMs), such as BERT (Devlin et al., 2019a), GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023b), LLaMA-2 (Touvron et al., 2023b), and Claude (AnthropicAI, 2023), have demonstrated impressive performance across a wide range of natural language processing (NLP) tasks. Major technology companies, such as Microsoft, Google, and Baidu, have deployed LLMs in their commercial products and services to enhance functionality. For instance, Microsoft leverages GPT-3.5 to improve search relevance ranking in new Bing (Mehdi, 2023). Since LLMs are notoriously complex "black-box" systems, their inner working mechanisms are opaque, and the high complexity makes model interpretation much challenging. This lack of model transparency can lead to the generation of harmful content or hallucinations in some cases (Weidinger et al., 2021). Therefore, it is critical to develop explainability to shed light on how these powerful models work.

Explainability[1] refers to the ability to explain or present the behavior of models in human-understandable terms (Doshi-Velez & Kim, 2017; Du et al., 2019a). Improving the explainability of LLMs is crucial for two key reasons. First, for general end users, explainability builds appropriate trust by elucidating the reasoning

[1] In the following sections, we use explainability and interpretability interchangeably.

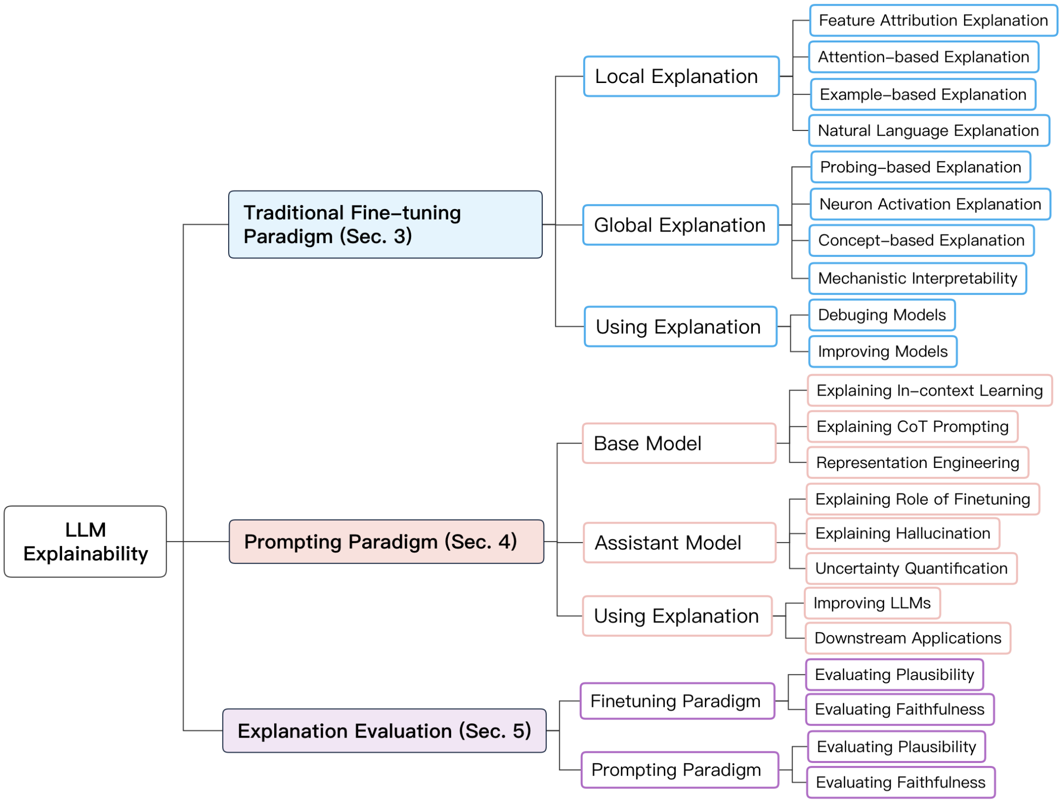

LLM Explainability
Traditional Fine–tuning Paradigm (Sec. 3)
Local Explanation
Feature Attribution Explanation
Attention–based Explanation
Example–based Explanation
Natural Language Explanation
Global Explanation
Probing–based Explanation
Neuron Activation Explanation
Concept–based Explanation
Mechanistic Interpretability
Using Explanation
Debuging Models
Improving Models
Prompting Paradigm (Sec. 4)
Base Model
Explaining In–context Learning
Explaining CoT Prompting
Representation Engineering
Assistant Model
Explaining Role of Finetuning
Explaining Hallucination
Uncertainty Quantification
Using Explanation
Improving LLMs
Downstream Applications
Explanation Evaluation (Sec. 5)
Finetuning Paradigm
Evaluating Plausibility
Evaluating Faithfulness
Prompting Paradigm
Evaluating Plausibility
Evaluating Faithfulness

Figure 1: We categorize LLM explainability into two major paradigms. Based on this categorization, we summarize different kinds of explainability techniques associated with LLMs belonging to these two paradigms. We also discuss evaluations for the generated explanations under the two paradigms.

mechanism behind model predictions in an understandable manner, without requiring technical expertise. With that, end users are able to understand the capabilities, limitations, and potential flaws of LLMs. Second, for researchers and developers, explaining model behaviors provides insight to identify unintended biases, risks, and areas for performance improvements. In other words, explainability acts as a debugging aid to quickly advance model performance on downstream tasks (Strobelt et al., 2018; Bastings et al., 2022; Yuksekgonul et al., 2023). It facilitates the ability to track model capabilities over time, make comparisons between different models, and develop reliable, ethical, and safe models for real-world deployment.

In contrast to traditional deep learning models, the scale of LLMs in terms of parameters and training data introduces both complex challenges and exciting opportunities for explainability research. Firstly, as models become larger, understanding and interpreting their decision-making processes grows more difficult due to increased internal complexity and the vastness of training data. This complexity also necessitates significant computational resources to generate explanations. On the one hand, traditionally practical feature attribution techniques, such as gradient-based methods (Sundararajan et al., 2017) and SHAP values (Lundberg & Lee, 2017a), could demand substantial computational power to explain LLMs with billions of parameters. This makes these explanation techniques less practical for real-world applications that end-users can utilize. On the other hand, this increased complexity makes in-depth analysis challenging, obstructing debugging and diagnosing of the models. Furthermore, comprehending the unique capabilities of LLMs in in-context learning (Li et al., 2023b) and chain-of-thought prompting (Wu et al., 2023a), as well as the phenomenon of hallucination, is indispensable to explain and improve models. Secondly, this scaling also spurs innovation in interpretability techniques and offers richer insights into model behavior. For instance, LLMs could provide chain-of-thought explanations for their own decision-making processes. Additionally, recent research finds LLMs can serve as tools to provide post-hoc explanations for predictions made by other machine learning models (Kroeger et al., 2023). To better understand and enhance LLMs, it is imperative to review available explainability techniques and develop an understanding of potential future directions.

In this paper, we provide a comprehensive overview of methods for interpreting Transformer-based language models. In Section 2, we introduce the two main paradigms in applying LLMs: 1) the traditional downstream fine-tuning paradigm and 2) the prompting paradigm. Based on this categorization, we review explainability methods for fine-tuned LLMs in Section 3, and prompted LLMs in Section 4. In Section 5, we discuss the evaluation of explainability methods. Finally, in Section 6, we further discuss research challenges in explaining LLMs compared to traditional deep learning models and provide insight on potential future research directions. This paper aims to comprehensively organize recent research progress on interpreting complex language models.

2 Training Paradigms of LLMs

The training of LLMs can be broadly categorized into two paradigms, *traditional fine-tuning* and *prompting*, based on how they are used for adapting to downstream tasks. Due to the substantial distinctions between the two paradigms, various types of explanations have been proposed respectively (shown in Figure 1).

2.1 Traditional Fine-Tuning Paradigm

In this paradigm, a language model is first pre-trained on a large corpus of unlabeled text data, and then fine-tuned on a set of labeled data from a specific downstream domain, such as SST-2, MNLI, and QQP on the GLUE benchmark (Wang et al., 2019). During fine-tuning, it is easy to add fully connected layers above the final encoder layer of the language model, allowing it to adapt to various downstream tasks (Rogers et al., 2021). This paradigm has shown success for medium-sized language models, typically containing up to one billion parameters. Examples include BERT (Devlin et al., 2019a), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), DeBERTa (He et al., 2021), etc. Explanations on this paradigm focus on two key areas: 1) Understanding how the self-supervised pre-training enables models to acquire a foundational understanding of language (e.g., syntax, semantics, and contextual relationships); and 2) Analyzing how the fine-tuning process equips these pre-trained models with the capability to effectively solve downstream tasks.

2.2 Prompting Paradigm

The prompting paradigm involves using prompts, such as natural language sentences with blanks for the model to fill in, to enable zero-shot or few-shot learning without requiring additional training data. Models under this paradigm can be categorized into two types, based on their development stages:

Base Model: As LLMs scale up in size and training data, they exhibit impressive new capabilities without requiring additional training data. One such capability is few-shot learning through prompting. This type of paradigm usually works on huge-size language models (with billions of parameters) such as GPT-3 (Brown et al., 2020), OPT (Zhang et al., 2022b), LLaMA-1 (Touvron et al., 2023a), LLaMA-2 (Touvron et al., 2023b), Falcon (Almazrouei et al., 2023). These models are called *foundation models or base models*[2], which can chat with users without further alignment with human preferences. Large-scale models typically fit into this paradigm, with size exceeding 1B. For example, LLaMA-2 (Touvron et al., 2023b) has up to 70B parameters. Explanations for base models aim to understand how models learn to leverage its pre-trained knowledge in response to the prompts.

Assistant Model: There are two major limitations of the base models: 1) they cannot follow user instructions as the pre-training data contains few instruction-response examples, and 2) they tend to generate biased and toxic content (Carlini et al., 2023). To address these limitations, the base models are further fine-tuned with supervised fine-tuning (see Figure 2) to achieve human-level abilities, such as open-domain dialogue. The key idea is to align the model's responses with human feedback and preferences. The most typical way for this process is through instruction tuning via (prompts, response) demonstration pairs and Reinforcement Learning from Human Feedback (RLHF). Models are trained with natural language feedback

[2] In this work, we refer foundation models to very large scale models such as LLaMA-2.

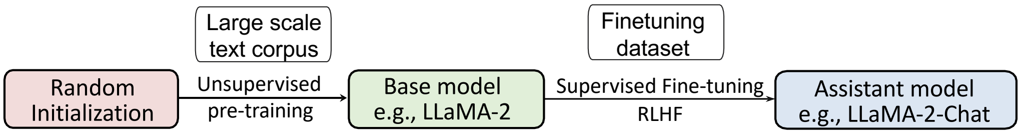

Large scale text corpus
Finetuning dataset
Random Initialization
Unsupervised pre-training
Base model e.g., LLaMA-2
Supervised Fine-tuning RLHF
Assistant model e.g., LLaMA-2-Chat

Figure 2: LLMs undergo unsupervised pre-training with random initialization to create a base model. The base model can then be fine-tuned through instruction tuning and RLHF to produce the assistant model.

to carry out complex, multi-turn conversations. Models belonging to this family include GPT-3.5 and GPT-4 (Bubeck et al., 2023) by OpenAI, Claude by Anthropic (AnthropicAI, 2023), and open-source models such as LLaMA-2-Chat by Meta (Touvron et al., 2023b), Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023). These models can be called *assistant models, chat assistants, or dialogue models.* Explanations here focus on understanding how models learn open-ended interactive behaviors from conversations.

3 Explanation for Traditional Fine-Tuning Paradigm

In this section, we review explanation techniques for LLMs trained with the pre-training and downstream fine-tuning paradigms. First, we introduce approaches to provide local explanations (Section 3.1) and global explanations (Section 3.2). Here, *local explanation* aims to provide an understanding of how a language model makes a prediction for a specific input instance, while *global explanation* aims to provide a broad understanding of how the LLM works overall. Next, we discuss how explanations can be used to debug and improve models (Section 3.3).

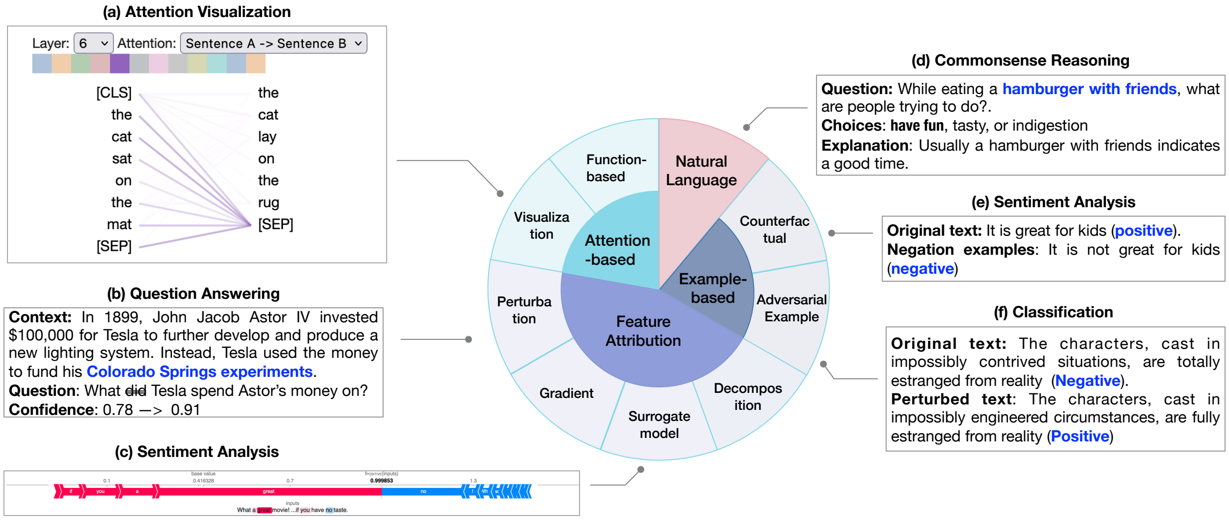
3.1 Local Explanation

The first category of explanations refers to explaining the predictions generated by LLM. Let us consider a scenario where we have a language model and we input a specific text into the model. The model then produces a classification output, such as sentiment classification or a prediction for the next token. In this scenario, the role of explanation is to clarify the process by which the model generated the particular classification or token prediction. Since the goal is to explain how the LLM makes the prediction for a specific input, we call it the *local explanation.* This category encompasses four main streams of approaches for generating explanations including feature attribution-based explanation, attention-based explanation, example-based explanation, and natural language explanation (see Figure 3).

3.1.1 Feature Attribution-Based Explanation

Feature attribution methods aim to measure the relevance of each input feature (e.g., words, phrases, text spans) to a model's prediction. Given an input text $\boldsymbol{x}$ comprised of n word features $\{x_1, x_2, ..., x_n\}$, a fine-tuned language model f generates an output $f(\boldsymbol{x})$. Attribution methods assign a relevance score $R(x_i)$ to the input word feature x_i to reflect its contribution to the model prediction $f(\boldsymbol{x})$. The methods that follow this strategy can be mainly categorized into four types: perturbation-based methods, gradient-based methods, surrogate models, and decomposition-based methods.

Perturbation-Based Explanation Perturbation-based methods work by perturbing input examples such as removing, masking, or altering input features and evaluating model output changes. The most straightforward strategy is *leave-one-out*, which perturbs inputs by removing features at various levels including embedding vectors, hidden units (Li et al., 2017), words (Li et al., 2016), tokens and spans (Wu et al., 2020b) to measure feature importance. The basic idea is to remove the minimum set of inputs to change the model prediction. The set of inputs is selected with a variety of metrics such as confidence score or reinforcement learning. However, this removal strategy assumes that input features are independent and ignores correlations among them. Additionally, methods based on the confidence score can fail due to pathological behaviors of overconfident models (Feng et al., 2018). For example, models can maintain high-confidence predictions even when the reduced inputs are nonsensical. This overconfidence issue can be mitigated via



Context: In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his **Colorado Springs experiments**.
Question: What ~~did~~ Tesla spend Astor's money on?
Confidence: 0.78 —> 0.91

Question: While eating a **hamburger with friends**, what are people trying to do?.
Choices: have fun, tasty, or indigestion
Explanation: Usually a hamburger with friends indicates a good time.

Original text: It is great for kids (**positive**).
Negation examples: It is not great for kids (**negative**)

Original text: The characters, cast in impossibly contrived situations, are totally estranged from reality (**Negative**).
Perturbed text: The characters, cast in impossibly engineered circumstances, are fully estranged from reality (**Positive**)

Figure 3: Local explanation is composed of four subareas. The organization of each subarea and examples for certain individual explanation methodology have been given. (a) Bipartite graph attention representation for attention matrix between sentence A and sentence B at the 6th layer (Vig, 2019); (b) Perturb the question by deleting "did", the confidence of the answer "Colorado Springs experiments" has even increased for the reduced question while the answer is nonsense for human (Feng et al., 2018); (c) Shapley values for transformer-based language models (Chen et al., 2023c); (d) Provide explanation to the important components of input text to assist in commonsense reasoning (Rajani et al., 2019); (e) Provide negative examples of input text to test model's ability in sentiment prediction and can also be used to improve model performance (Wu et al., 2021); (f) Change the input text in an imperceptible way for humans but the classification is distracted from the original (Jin et al., 2020).

regularization with regular examples, label smoothing, and fine-tuning models' confidence (Feng et al., 2018). Besides, current perturbation methods tend to generate out-of-distribution data. This can be alleviated by constraining the perturbed data to remain close to the original data distribution (Qiu et al., 2021).

Gradient-Based Explanation Gradient-based attribution techniques determine the importance of each input feature by analyzing the partial derivatives of the output with respect to each input dimension. The magnitude of derivatives reflects the sensitivity of the output to changes in the input. The basic formulation of raw gradient methods is described as $\boldsymbol{s}_j = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_j}$, where $f(\boldsymbol{x})$ is the prediction function of the network and $\boldsymbol{x}_j$ denotes the input vector. This scheme has also been improved as gradient × input (Kindermans et al., 2017) and has been used in various explanation tasks, such as computing the token-level attribution score (Mohebbi et al., 2021). However, vanilla gradient-based methods have some major limitations. First, they do not satisfy the input invariance, meaning that input transformations such as constant shift can generate misleading attributions without affecting the model prediction (Kindermans et al., 2017). Second, they fail to deal with zero-valued inputs. Third, they suffer from gradient saturation where large gradients dominate and obscure smaller gradients. The difference-from-reference approaches, such as integrated gradients (IG), are believed to be a good fit to solve these challenges by satisfying more axioms for attributions (Sundararajan et al., 2017). The fundamental mechanism of IG and its variants is to accumulate the gradients obtained as the input is interpolated between a reference point and the actual input. The baseline reference point is critical for reliable evaluation, but the criteria for choosing an appropriate baseline remain unclear. Some use noise or synthetic reference with training data, but performance cannot be guaranteed (Lundstrom et al., 2022). In addition, IG struggles to capture output changes in saturated regions and should focus on unsaturated regions (Miglani et al., 2020). Another challenge of IG is the computational overhead to achieve high-quality integrals. Since IG integrates along a straight line path that does not fit well the discrete word embedding space, variants have been developed to adapt it for language models (Sikdar et al., 2021; Sanyal & Ren, 2021; Enguehard, 2023).

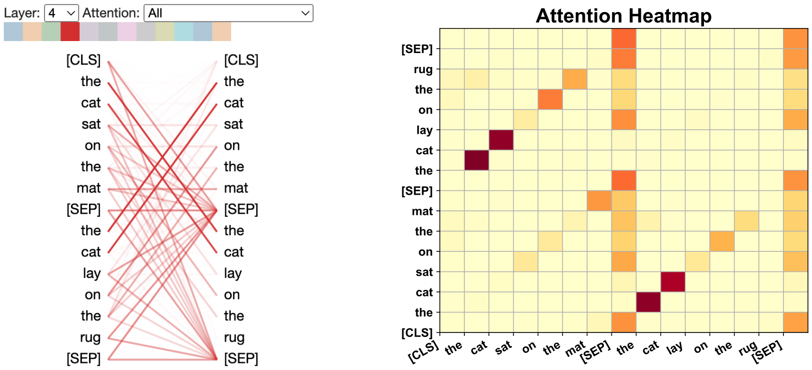
Surrogate Models Surrogate models methods use simpler, more human-comprehensible models to explain individual predictions of black-box models. These surrogate models include decision trees, linear models, decision rules, and other white-box models that are inherently more understandable to humans. The explanation models need to satisfy additivity, meaning that the total impact of the prediction should equal the sum of the individual impacts of each explanatory factor. Also, the choice of interpretable representations matters. Unlike raw features, these representations should be powerful enough to generate explanations yet still understandable and meaningful to human beings. An early representative local explanation method called LIME (Ribeiro et al., 2016) employs this paradigm. To generate explanations for a specific instance, the surrogate model is trained on data sampled locally around that instance to approximate the behavior of the original complex model in the local region. However, it is shown that LIME does not satisfy some properties of additive attribution, such as local accuracy, consistency, and missingness (Lundberg & Lee, 2017b). SHAP is another framework that satisfies the desirable properties of additive attribution methods (Lundberg & Lee, 2017b). It treats features as players in a cooperative prediction game and assigns each subset of features a value reflecting their contribution to the model prediction. Instead of building a local explanation model per instance, SHAP computes Shapley values (Shapley et al., 1953) using the entire dataset. Challenges in applying SHAP include choosing appropriate methods for removing features and efficiently estimating Shapley values. Feature removal can be done by replacing values with baselines like zeros, means, or samples from a distribution, but it is unclear how to pick the right baseline. Estimating Shapley values also faces computational complexity exponential in the number of features. Approximation strategies including weighted linear regression, permutation, and other model-specific methods have been adopted (Chen et al., 2023c) to estimate Shapley values. Despite complexity, SHAP remains popular and widely used due to its expressiveness for large deep models. To adapt SHAP to Transformer-based language models, approaches such as TransSHAP have been proposed (Chen et al., 2023c; Kokalj et al., 2021). TransSHAP mainly focuses on adapting SHAP to sub-word text input and providing sequential visualization explanations that are well suited for understanding how LLMs make predictions.

Decomposition-Based Methods Decomposition techniques aim to break down the relevance score into linear contributions from the input. Some work assign relevance score directly from the final output layer to the input (Du et al., 2019b). The other line of work attributes relevance score layer by layer from the final output layer toward the input. Layer-wise relevance propagation (LRP) (Montavon et al., 2019) and Taylor-type decomposition approaches (DTD) (Montavon et al., 2015) are two classes of commonly used methods. The general idea is to decompose the relevance score $R_j^{(l+1)}$ of neuron j in layer $l+1$ to each of its input neuron i in layer l, which can be formulated as: $R_j^{(l+1)} = \sum_i R_{i \leftarrow j}^{(l,l+1)}$. The key difference is in the relevance propagation rules used by LRP versus DTD. These methods can be applied to break down relevance scores into contributions from model components such as attention heads (Voita et al., 2019), tokens, and neuron activation (Voita et al., 2021). Both methods have been applied to derive the relevance score of inputs in Transformer-based models (Wu & Ong, 2021; Chefer et al., 2021).

3.1.2 Attention-Based Explanation

Attention mechanism is often viewed as a way to attend to the most relevant part of inputs. Intuitively, attention may capture meaningful correlations between intermediate states of input that can explain the model's predictions. Many existing approaches try to explain models solely based on the attention weights or by analyzing the knowledge encoded in the attention. These explanation techniques can be categorized into three main groups: visualization methods, function-based methods, and probing-based methods. As probing-based techniques are usually employed to learn global explanations, they are discussed in Section 3.2.1. In addition, there is an extensive debate in research on whether attention weights are actually suitable for explanations. This topic will be covered later in the discussion.

Visualizations Visualizing attentions provides an intuitive way to understand how models work by showing attention patterns and statistics. Common techniques involve visualizing attention heads for a single



(a) Bipartite Graph

(b) Heatmap

Figure 4: Bipartite graph attention representation and heatmap for attention matrix.

input using bipartite graphs or heatmaps. These two methods are simply disparate visual representation of attentions, one as a graph and the other as a matrix, as illustrated in Figure 4. Visualization systems differ in their ability to show relationships at multiple scales, by representing attention in various forms for different models. At the input data level, attention scores for each word/token/sentence pairs between the premise sentence and the assumption sentence are shown to evaluate the faithfulness of the model prediction (Vig, 2019). Some systems also allow users to manually modify attention weights to observe effects (Jaunet et al., 2021). At the neuron level, individual attention heads can be inspected to understand model behaviors (Park et al., 2019; Vig, 2019; Hoover et al., 2020; Jaunet et al., 2021). At the model level, attention across heads and layers is visualized to identify patterns (Park et al., 2019; Vig, 2019; Yeh et al., 2023). One notable work focuses on visualizing attention flow to trace the evolution of attention, which can be used to understand information transformation and enable training stage comparison between models (DeRose et al., 2020). Thus, attention visualization provides an explicit, interactive way to diagnose bias, errors, and evaluate decision rules. Interestingly, it also facilitates formulating explanatory hypotheses.

Function-Based methods Since raw attention is insufficient to fully explain model predictions, some studies have developed enhanced variants as replacements to identify important attributions for explanation. Gradient is a well-recognized metric for measuring sensitivity and salience, so it is widely incorporated into self-defined attribution scores. These self-designed attribution scores differ in how they define gradients involving attention weights. For example, gradients can be partial derivatives of outputs with respect to attention weights (Barkan et al., 2021) or integrated versions of partial gradients (Hao et al., 2021). The operations between gradients and attention can also vary, such as element-wise products. Overall, these attribution scores that blend attention and gradients generally perform better than using either alone, as they fuse more information that helps to highlight important features and understand networks.

Debate Over Attention There is extensive research evaluating attention heads, but the debate over the validity of this approach is unlikely to be resolved soon. The debate stems from several key aspects. First, some works compare attention-based explanations with those from other methods like LIME. They find that attention often does not identify the most important features for prediction (Serrano & Smith, 2019; Jain & Wallace, 2019). They provide inferior explanations compared to these alternatives (Thorne et al., 2019) or cannot be correlated with other explanation methods (Jain & Wallace, 2019; Liu et al., 2020; Ethayarajh & Jurafsky, 2021). Second, some directly criticize the usefulness of the attention mechanism in model predictions. They argue that raw attention fails to capture syntactic structures in text and may not contribute to predictions as commonly assumed (Mohankumar et al., 2020). In addition, raw attention contains redundant information that reduces its reliability in explanations (Bai et al., 2021; Brunner et al., 2019). However, other studies contradict these claims. For example, evaluating explanation models for consistency can pose challenges across various approaches, not limited to attention alone (Neely et al., 2021). Besides, manipulation of attention weights without retraining can bias evaluations (Wiegreffe &

Pinter, 2019). Furthermore, attention heads in BERT have been shown to encode syntax effectively (Clark et al., 2019). To make attention explainable, technical solutions have also been explored by optimizing input representation (Mohankumar et al., 2020), regularizing learning objectives (Moradi et al., 2021), avoiding biased learning (Bai et al., 2021) and even incorporating human rationales (Arous et al., 2021). But the core reason for the ongoing debates is the lack of well-established evaluation criteria, which will be further discussed in Section 5.1.

3.1.3 Example-Based Explanations

Example-based explanations intend to explain model behavior from the perspective of individual instances (Koh & Liang, 2017). Unlike model-based or feature-based explanations, example-based explanations illustrate how a model's output changes with different inputs. We focus on adversarial examples, counterfactual explanations, and data influence. Adversarial examples are generally synthesized by manipulating less important components of input data. They reveal cases where the model falters or errs, illuminating its weaknesses. In contrast, counterfactual explanations are generated mostly by changing significant parts of input data, and they are popular in scenarios like algorithmic recourse, as providing remedies to a desirable outcome. Unlike manipulating inputs, data influence examines how training data impacts a model's predictions on testing data.

Adversarial Example Studies show that neural models are highly susceptible to small changes in the input data. These carefully crafted modifications can alter model decisions while barely being noticeable to humans. Adversarial examples are critical in exposing areas where models fail and are usually added to training data to improve robustness and accuracy. Adversarial examples are initially generated by word-level manipulations such as errors, removal, and insertion, which are obvious upon inspection. More advanced token-level perturbation methods like TextFooler (Jin et al., 2020) have been advanced, which strategically target important words first based on ranking. A candidate word is then chosen based on word embedding similarity, same part-of-speech, sentence semantic similarity, and prediction shift. However, word embedding is limited in sentence representation compared to contextualized representations, often resulting in incoherent pieces. By focusing on contextualized representations, a range of work adopting the mask-then-infill procedure has achieved state-of-the-art performance (Garg & Ramakrishnan, 2020; Li et al., 2021b). They leverage pre-trained masked language models like BERT for perturbations including replacement, insertion, and merging. Typically, a large corpus is employed to train masked language models, generate contextualized representations and obtain token importance. Then models are frozen and perturbation operations are performed on tokens in a ranked order. For replacement, the generated example replaces the masked token. For injection, the new token is inserted into the left or right of the masked token. For merging, a bigram is masked and replaced with one token. SemAttack (Wang et al., 2022b) proposes a more general and effective framework applicable to various embedding spaces including typo space, knowledge space, and contextualized semantic space. The input tokens are first transformed into an embedding space to generate perturbed embeddings that are iteratively optimized to meet attack goals. The experiment shows that replacing 5% of words reduces BERT's accuracy from 70.6% to 2.4% even with defenses in a white-box setting. SemAttack's outstanding attack performance might because it directly manipulates embeddings.

Counterfactual Explanation Counterfactual explanation is a common form of casual explanation, treating the input as the cause of the prediction under the Granger causality. Given an observed input $\boldsymbol{x}$ and a perturbed $\hat{\boldsymbol{x}}$ with certain features changed, the prediction $\boldsymbol{y}$ would change to $\hat{\boldsymbol{y}}$. Counterfactual explanation reveals what would have happened based on certain observed input changes. They are often generated to meet up certain needs such as algorithmic recourse by selecting specific counterfactuals. Examples can be generated by humans or perturbation techniques like paraphrasing or word replacement. A representative generator, Polyjuice (Wu et al., 2021), supports multiple permutation types for input sentences, such as deletion, negation and shuffling. It can also perturb tokens based on their importance. Polyjuice then finetunes GPT-2 on specific pairs of original and perturbed sentences tailored to downstream tasks, to provide realistic counterfactuals. It generates more extensive counterfactuals with a median speed of 10 seconds per counterfactual, compared to 2 minutes for previous crowd workers-dependent methods (Kaushik et al., 2020). Counterfactual explanation generation has been framed as a two-stage approach involving first masking/selecting important tokens and then infilling/editing those tokens (Treviso et al., 2023; Ross et al., 2021). Specifically, MiCE uses gradient-based attribution to select tokens to mask in the first stage and focuses on optimizing for minimal edits through binary search (Ross et al., 2021). In contrast, CREST leverages rationales from a selective rationalization model and relaxes this hard minimality constraint of MiCE. Instead, CREST uses the sparsity budget of the rationalizer to control closeness (Treviso et al., 2023). Experiments show that both methods generate high-quality counterfactuals in terms of validity and fluency.

Data Influence This family of approaches characterize the influence of individual training sample by measuring how much they affect the loss on test points (Yeh et al., 2018). The concept originally came from statistics, where it describes how model parameters are affected after removing a particular data point. By observing patterns of influence, we can deepen our understanding of how models make predictions based on their training data. Since researchers have come to recognize the importance of data, several methods have been developed to analyze models from a data-centric perspective. Firstly, influence functions enable us to approximate the concept by measuring loss changes via gradients and Hessian-vector products without the necessity of retraining the model (Koh & Liang, 2017). Yeh et al. (2018) decompose a prediction of a test point into a linear combination of training points, where positive values denote excitatory training points and negative values indicate inhibitory points. Data Shapley employs Monte Carlo and gradient-based methods to quantify the contribution of data points to the predictor performance, and the higher Shapley value tells the desired data type to improve the predictor (Ghorbani & Zou, 2019). Another method uses stochastic gradient descent (SGD) and infers the influence of a training point by analyzing minibatches without that point using the Hessian vector of the model parameters (Hara et al., 2019). Based on such an approach, TracIn derives the influence of training points using the calculus theorem with checkpoints during the training process (Pruthi et al., 2020). However, the aforementioned methods often come with an expensive computational cost even when applied to a medium-sized model. To address this, two key dimensions can be considered: 1) reducing the search space and 2) decreasing the number of approximated parameters in the Hessian vector. Guo et al. (2020) also demonstrates the applicability of the influence function in model debugging. Recently, Anthropic has employed the Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC) approximation to scale this method to LLMs with 810 million, 6.4 billion, 22 billion, and 52 billion parameters. The result indicates that as model scale increases, influential sequences are better at capturing the reasoning process for queries, whereas smaller models often provide semantically unrelated pieces of information (Grosse et al., 2023).

3.1.4 Natural Language Explanation

Natural language explanation in NLP refers to explaining a model's decision-making on an input sequence with generated text. The basic approach for generating natural language explanations involves training a language model using both original textual data and human-annotated explanations. The trained language model can then automatically generate explanations in natural language (Rajani et al., 2019). As explanations provide additional contextual space, they can improve downstream prediction accuracy and perform as a data augmentation technique (Luo et al., 2022; Yordanov et al., 2022). Apart from the explain-then-predict approach, predict-then-explain and joint predict-explain methods have also been investigated. The choice of methods depends on the purpose of the task. But the reliability of applying generated explanations still necessitates further investigation. It is worth noting that both the techniques introduced in this section and the CoT explanations covered later in Section 4 produce natural language explanations. However, the explanations covered here are typically generated by a separate model, while CoT explanations are produced by the LLMs themselves.

3.2 Global Explanation

Different from local explanations that aim to explain a model's individual predictions, global explanations offer insights into the inner workings of language models. Global explanations aim to understand what the individual components (neurons, hidden layers, and larger modules) have encoded and explain the knowledge/linguistic properties learned by the individual components. We examine three main approaches for global explanations: probing methods that analyze model representations and parameters, neuron activation analysis to determine model responsiveness to input, and concept-based methods.

3.2.1 Probing-Based Explanations

The self-supervised pre-training process leads to models that acquire broad linguistic knowledge from the training data. The probing technique refers to methods used to understand the knowledge that LLMs such as BERT have captured.

Classifier-Based Probing The basic idea behind classifier-based probing is to train a shallow classifier on top of the pre-trained or fine-tuned language models such as BERT (Devlin et al., 2019b), T5 (Raffel et al., 2020). To perform probing, the parameters of the pre-trained models are first frozen, and the model generates representations for input words, phrases, or sentences and learns parameters like attention weights. These representations and model parameters are fed into a *probe* classifier, whose task is to identify certain linguistic properties or reasoning abilities acquired by the model. Once the probe is trained, it will be evaluated on a holdout dataset. The labeled data comes from available taggers or gold-annotated datasets. Although each probe classifier is often tailored for a certain task, the scheme for training classifiers to probe different knowledge is consistent. Related studies will be presented according to probed model components, i.e. vector representations and model parameters.

We first examine works that study *vector representations* to measure embedded knowledge. In this category, knowledge means either *syntax knowledge* at a low level or *semantic knowledge* at a high level. Studies demonstrate that lower layers are more predictive of word-level syntax, whereas higher layers are more capable of capturing sentence-level syntax and semantic knowledge (Belinkov et al., 2017; Peters et al., 2018; Blevins et al., 2018; Jawahar et al., 2019).

Syntactic labels can be further categorized into word- or sentence-level categories. The word-level syntactic labels provide information about each individual word, such as part-of-speech tags, morphological tags, smallest phrase constituent tags, etc. The sentence-level syntactic labels describe attributes of the whole sentence, such as voice (active or passive), tense (past, present, future), and top-level syntactic sequence. For word-level syntax probing, parse trees are often introduced by dependency parser (Dozat & Manning, 2017) to help extract dependency relations (Tenney et al., 2019b). A structural probe is also developed to identify parse trees in a specific vector space by measuring the syntactic distance between all pairs of words with distance metrics (Hewitt & Manning, 2019; Chen et al., 2021). This demonstrates that syntactic knowledge is embedded in vector representations and is popular for reconstructing dependency trees for probing tasks. However, concerns emerged about whether probing classifiers learn syntax in representations or just the task. Some believe that only rich syntax representations enable simple classifiers to perform well (Lin et al., 2019). Kunz & Kuhlmann (2020) overthrow these claims by demonstrating that its good performance comes from encoding local neighboring words. A study shows that classifiers rely on semantic cues fail to extract syntax (Maudslay & Cotterell, 2021). In contrast, other research reveals that models such as BERT encode the corresponding information in a variety of ways (Mohebbi et al., 2021; Li et al., 2021a). Therefore, the validity of probing syntactic information still requires further investigation. Since sentence-level syntactic information is generally distributed in each word, the prediction for them is simpler with probing classifier without dependency tree retrieval. Local syntax and semantics are usually studied together as they investigate the same objects such as neurons, layers, and contextual representation. The differences are mainly due to their training objectives and training data (Tenney et al., 2019a).

The ability to learn semantic knowledge is often examined on tasks like coreference resolution, named entity labeling, relation classification, question types classification and supporting facts, etc. (Van Aken et al., 2019). A prominent framework called edge probing (Tenney et al., 2019b) has been advanced to provide exhaustive syntactic and semantic probing tools. Differently, it takes both pre-trained representation and integer spans as input and transforms them into fixed-length span representations that are fed to train a probing classifier. Because of the definition of span representation, such approach becomes extremely versatile and widely applied in syntactic and semantic probing tasks. Some works simply probe referential relations by measuring the similarity between transformed representation of pronouns and preceding words within a fixed length, and assigning a higher probability to more similar ones (Sorodoc et al., 2020). Probing work involving prompts usually faces challenges with zero-shot and few-shot learning. The evaluation with these models is more complicated as prompt quality also significantly influences performance (Zhang et al., 2022a). Even with carefully designed datasets and prompt design, the result still needs further examination.

On the other hand, probing classifiers for *attention heads* are designed in a similar fashion where a shallow classifier is trained on top of pre-trained models to predict certain features. Apart from relating attention heads to syntax and semantics, patterns in attention heads are also studied. A representative work trains classifiers to identify patterns using self-attention maps sampled on random inputs, then prunes heads based on this to improve model efficiency (Kovaleva et al., 2019; Clark et al., 2019). Instead of making predictions, some work regards attention as semantic information indicators and traces it backward through layers, accumulates it and distributes it to input tokens to denote semantic information (Wu et al., 2020a). But the question is whether traced attention equivalently represents semantic information across different heads.

Although high probing performance is often attributed to the quality and interpretability of representations (Belinkov, 2022), this assumption remains largely unproven and difficult to validate. Before we can comprehensively address such challenges, adding constraints like selectivity (Hewitt & Liang, 2019), which measures how selectively a probe targets the linguistic property of interest compared to an unrelated control task, may help mitigate potential probe biases in the interim.

Parameter-Free Probing There is another branch of data-centric probing techniques that does not require probing classifiers. Instead, they design datasets tailored to specific linguistic properties like grammar (Marvin & Linzen, 2018). The encoding model's performance illustrates its capability in capturing those properties. For language models, the measurement is based on whether the probability of positive examples is higher than that of negative examples.

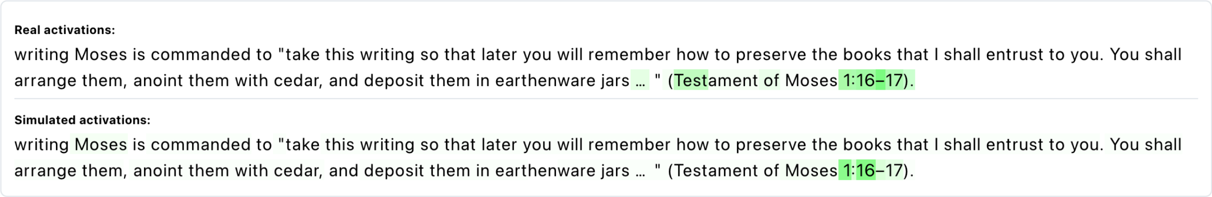
Probing tasks can also be performed with data-driven prompt search, where certain knowledge are examined via language model's text generation or completion abilities (Petroni et al., 2019; Apidianaki & Soler, 2021; Li et al., 2022). For instance, Ravichander et al. (2020) proves that BERT encodes hypernymy information by completing cloze tasks i.e. filling blanks in incomplete sentences in zero-shot settings. And the result demonstrates that BERT performs well in providing the right answer in the top 5 for all samples. However, as argued by (Zhong et al., 2021), training datasets include regularities that prompting methods can exploit to make predictions. The real factual knowledge captured by language models becomes obscure.

3.2.2 Neuron Activation Explanation

Instead of examining the whole vector space, neuron analysis looks into individual dimensions, i.e. neurons in representations, that are crucial for model performance or associated with specific linguistic properties.

One simple line of work follows two main steps: first, identifying important neurons in an unsupervised manner. Second, learning relations between linguistic properties and individual neurons in supervised tasks. On the assumption that different models learning similar properties usually share similar neurons, these shared neurons are ranked according to various metrics such as correlation measurements and learned weights (Bau et al., 2018; Dalvi et al., 2019). Alternatively, conventional supervised classification can also be adopted to find important neurons in a given model (Dalvi et al., 2019). The importance of ranked neurons is verified quantitatively via ablation experiments, e.g., masking, erasure, visualization, etc. Other probing techniques like greedy Gaussian probing have also emerged to identify important neurons (Torroba Hennigen et al., 2020). However, existing methods struggle to balance accuracy and selectivity (Antverg & Belinkov, 2022).

Intuitively, all neurons should be examined to make explanation. However, due to the expensive computational cost and the claim that only a small subset of neurons are important for making decisions (Bau et al., 2018; Antverg & Belinkov, 2022) , existing approaches are always integrated with ranking algorithms. With the increasing generalization capabilities of LLMs, it becomes feasible to provide explanations for individual neurons. A recent study by OpenAI demonstrates the use of GPT-4 to generate natural language explanations for individual neuron activation in GPT-2 XL (OpenAI, 2023a). It uses GPT-4 to summarize the patterns in text that trigger high activation values for a given GPT-2 XL neuron. For example, GPT-4 could summarize the pattern of a neuron as: *references to movies, characters, and entertainment.* The quality of each neuron explanation is evaluated by testing how well it allows GPT-4 to simulate the



Real activations:

writing Moses is commanded to "take this writing so that later you will remember how to preserve the books that I shall entrust to you. You shall arrange them, anoint them with cedar, and deposit them in earthenware jars … " (Testament of Moses 1:16–17).

Simulated activations:

writing Moses is commanded to "take this writing so that later you will remember how to preserve the books that I shall entrust to you. You shall arrange them, anoint them with cedar, and deposit them in earthenware jars … " (Testament of Moses 1:16–17).

Figure 5: Activation visualization of the 131st neuron in the 5th layer of the GPT-2. The simulated explanation from GPT-4 indicates that the 131st neuron in the fifth layer of GPT-2 is activated by citations. The real activation of this neuron confirms the accuracy of the simulated explanation provided by GPT-4.

real neuron's behavior on new text examples. Explanations are scored based on the correlation between GPT-4's simulated activation and the true activation (see Figure 5). High correlation indicates an accurate explanation that captures the essence of what the neuron encodes. More than 1,000 GPT-2 XL neurons were found to have high-scoring explanations from GPT-4, which accounts for most of their behavior. This auto-generated natural language provides intuitive insight into the emergent inner computations and feature representations in GPT-2 XL. A common limitation in explaining individual components of LLMs is the lack of ground-truth explanation annotations for individual components. Without these annotations, the evaluation of component-level explanations remains challenging.

Another recent study proposes the Summarize and Score (SASC) explanation pipeline to generate natural language explanations to explain modules from large language models (Singh et al., 2023). First, SASC generates candidate explanations using a pre-trained language model to find n-grams that elicit the most positive response from the module f. The SASC then evaluates each candidate explanation by generating synthetic data based on the explanation and testing how f responds to those data. The authors apply SASC to explain modules within BERT (bert-base-uncased), which is then compared with human-labeled explanations. The comparison indicates that SASC explanations are sometimes similar to human explanations.

3.2.3 Concept-Based Explanation

Concept-based interpretability algorithms map the inputs to a sets of concepts and measure importance score of each pre-defined concept to model predictions. By introducing abstract concepts, models can be explained in a human-understandable fashion rather than on low-level features. Information in latent space can also be transformed into comprehensible explanations. A representative framework named TCAV (Kim et al., 2018) uses directional derivatives to quantify the contribution of defined concepts to the model predictions. It first represents the concept with a set of examples and then learns a linear classifier as "concept activation vector"(CAV) to detect interested concept. The learned vector is used as input changes in the direction of concepts to measure prediction sensitivity with concepts i.e. importance score of concepts. Originally proposed for computer vision, TCAV has also been tailored to NLP models for sentiment classification using the IMDB sensitivity dataset (Captum). Specifically, two concepts were explored: Positive Adjectives and Neutral. The Positive Adjectives concept refers to a group of adjectives that express positive feelings. The Neutral concept spans broader domains and is distinct from Positive Adjectives. For sentences with negative sentiment, the TCAV scores indicate the Positive Adjectives score is relatively low compared to Neutral, which is consistent with human understanding. However, TCAV requires additional data to describe concepts and performance of the concept classifier can hardly be guaranteed. An alternative way of selecting concepts is to identify those learned by neurons through probing tasks with annotated datasets (Mu & Andreas, 2021). The study shows that neurons produce explanations not only based on individual concepts but also compositions of logical forms. The more neurons are interpretable, the more accurate the model is. A common pitfall of concept-based explanations is how to define useful concepts. Besides, it is always constrained by available descriptive datasets.

3.2.4 Mechanistic Interpretability

Mechanistic interpretability understands language models by investigating individual neurons and especially their connections in terms of *circuits* (Anthropic, 2023; Bricken et al., 2023). Due to the motivation to regard parts of neural models as functional components, we discuss this line of work separately.

Circuits was originally proposed to explain vision models that are intuitive to comprehend, where detectors for complex objects can be built out of simple building blocks such as line detectors, curve detector, and so on. One stream of work studies the hidden representation of neural networks. These representations can be visualized with features. They believe that complex feature detectors can be implemented from earlier and easier feature detectors. Besides, different features can be spread across many polysemantic neurons also known as superposition (Olah et al., 2020a). Another line of work studies weights that connect neurons aiming to find meaningful algorithms that implement simple logic. They approach subgraphs of networks with circuits denoting linear combination of features as well as logical operations, which is crucial to establish a casual relationship for predictions. Built on top of neuron-level explanation in circuits, larger-scale functional components have also been explored. Three phenomena have been identified: 1) branch specialization, 2) weight banding, 3) equivariance. Branch specialization describes the feature organization between branches, where a given type of feature was observed to group into a branch. This phenomenon exists at different levels of layers, and the same branch specialization might be robust across different architectures and tasks (Voss et al., 2021). The weight banding usually appears in the final layer of vision models with global average pooling (Petrov et al., 2021). Equivariance captures symmetries in neural networks where many neurons are transformed from the basic version (Olah et al., 2020b).

When it comes to transformers, circuits usually work and are interpreted in a different way because of their architectures. One-layer and two-layer attention-only models have been investigated recently. For one-layer attention-only models, bigram and skip-trigram tables can be accessed from weights. However, two-layer attention-only transformers demonstrate "induction head" by composing attention heads from different layers (Elhage et al., 2021). The induction head consists of two attention heads. The first attention head is responsible for copying information from the previous token into the next token, while the second one is used to deduct the following token with information from the first attention head. Such mechanism is believed to be the source of in-context learning, which has been demonstrated with multiple less conclusive evidences. For example, the phase change observed on cooccurrence of in-context learning and induction heads, and corresponding in-context learning shifts after perturbing or knocking out induction heads. However, due to the complex components of state-of-art language models, such as multiple layers and multilayer perceptrons, it remains to be seen whether the theory of the "induction heads" in these models still holds (Olsson et al., 2022). Alternatively, some work focuses on feedforward layers that contain most of the information. Each key in transformers is taken as a memory of textual patterns in the training examples. The values induce output distribution based on keys (Geva et al., 2020). By tracing the casual effects of hidden state activation within GPT and altering model weights that are decisive at model predictions, a range of middle layers are identified as relevant with facts (Meng et al., 2022). Another study transfers the feedforward layer as a sub-update vector which is interpreted as a small set of human-interpretable concepts (Geva et al., 2022).

However, unlike digital circuits that have deterministic functions in each part, large-scale neural networks are more elastic and versatile in composition such as robust to remove entire layers (Veit et al., 2016; McGrath et al., 2023). In addition, most existing hypotheses have not been examined on large language models. Recently, Lieberum et al. (2023) explore scalability of circuit analysis in the 70B Chinchilla model. The result demonstrates activation patching (Meng et al., 2022), attention pattern visualization (Elhage et al., 2021) and logit attribution can be well adapted rather than correct letter heads that moving information from the correct content tokens to the final token (Lieberum et al., 2023). Therefore, the circuit-based explanation still requires further investigation on LLMs.

3.3 Making Use of Explanations

In the previous subsections, we introduced methods to generate explanations for LLMs. In this subsection, we discuss how explainability can be used as a tool to debug and improve models.

3.3.1 Debugging Models

Post-hoc explainability methods can be used to analyze model feature importance patterns to identify biases or limitations in its behavior (Du et al., 2023). For example, if the model consistently attends to certain tokens in the input sequence regardless of the context, this may indicate that the model relies on heuristics or biases rather than truly understanding the meaning of the input sequence. Recent work has used Integrated Gradients to debug trained language models in natural language understanding tasks, finding that they use shortcuts rather than complex reasoning for prediction (Du et al., 2021). Specifically, these models favor features from the head of long-tailed distributions, picking up these shortcut cues early in training. This shortcut learning harms model robustness and generalization to out-of-distribution samples. Integrated Gradient explanations have also been used to examine the adversarial robustness of language models (Chen & Ji, 2022). The explanations reveal that models robust to adversarial examples rely on similar features, while non-robust models rely on different key features. These insights have motivated the development of more robust adversarial training methods.

3.3.2 Improving Models

Regularization techniques can be used to improve the performance and reliability of model explanations. Specifically, explanation regularization (ER) methods aim to improve LLM generalization by aligning the model's machine rationales (which tokens it focuses on) with human rationales (Joshi et al., 2022). For example, a framework called AMPLIFY is proposed that generates automated rationales using post-hoc explanation methods (Ma et al., 2023). These automated rationales are feed as part of prompts to LLM for prediction. Experiments show that AMPLIFY improves the accuracy of LLMs by 10-25% for various tasks, even when human rationale is lacking. Another study proposes ER-TEST (Joshi et al., 2022), a framework that evaluates the out-of-distribution (OOD) generalization of ER models along three dimensions: unseen dataset tests, contrast set tests, and functional tests. This provides a more comprehensive evaluation than just in-distribution performance. They consider three types of explainability methods, including Input*Gradient, attention-based rationale (Stacey et al., 2022), and learned rationale (Chan et al., 2022a). Across sentiment analysis and natural language inference tasks/datasets, ER-TEST shows that ER has little impact on in-distribution performance but yields large OOD gains. An end-to-end framework called XMD was proposed for explanation-based debugging and improvement (Lee et al., 2022). XMD allows users to provide flexible feedback on task- or instance-level explanations via an intuitive interface. It then updates the model in real time by regularizing it to align explanations with user feedback. Using XMD has been shown to improve models' OOD performance on text classification by up to 18%.

4 Explanation for Prompting Paradigm

As language models scale up, prompting-based models exhibit emergent abilities that require new perspectives to elucidate their underlying mechanisms. However, the aggressive surge in model scale renders traditional explanation methods unsuitable. The challenges of applying certain explanability techniques targeting traditional fine-tuning paradigms to prompting-based paradigms can be summarized from multiple facets. For example, prompting-based models rely on reasoning abilities (Wei et al., 2023b), which makes localized or example-specific explanations much less meaningful. Additionally, computationally demanding explanation techniques quickly become infeasible at the scale of hundreds of billions of parameters or more. Further, the intricate inner workings and reasoning processes of prompting-based models are too complex to be effectively captured by simplified surrogate models.

In light of these challenges, new explanation techniques tailored to this prompting paradigm are emerging. For example, chain of thought (CoT) explanations may provide a more suitable approach for understanding and explaining the behaviors of large language models based on prompting. Besides, methods that focus on identifying influential examples that contribute to predictions are gaining importance. Identifying these pivotal data points may significantly enhance our understanding of dataset composition. Global explanation techniques for traditional fine-tuning paradigms are widely employed to prompting-based LLMs as well. Particularly these techniques that capable of delivering high-level explanations such as concept-based explanation and module-based explanation.

In this section, we first introduce techniques to explain models belonging to the prompting paradigm, including 1) explaining base models such as LLaMA-2 (Section 4.1), 2) explaining assistant models such as LLaMA-2-Chat (Section 4.2), and 3) how to harness the reasoning and explaining ability of LLMs to improve the predictive performance of language models and enable beneficial applications (Section 4.3).

4.1 Base Model Explanation

As the scale of language models increases, they exhibit new abilities like few-shot learning, i.e., the ability to learn concepts from just a few examples. They also demonstrate a chain-of-thought(CoT) prompting ability, which allows feeding a sequence of prompts to the model to steer its generation in a particular direction and have it explain its reasoning (Wei et al., 2022). Given these emerging properties, the explainability research has three main goals: 1) understanding how these large language models can grasp new tasks so quickly from limited examples, which helps end-users interpret the model's reasoning, 2) explaining CoT prompting, and 3) and representation engineering.

4.1.1 Explaining In-context Learning

Explainable AI techniques have been used to elucidate how prompting works in LLMs. Specifically, we discuss techniques that shed light on how in-context learning (ICL) influences model behavior.

One study uses the SST-2 sentiment analysis benchmark as a baseline task to explain the in-context learning paradigm (Li et al., 2023b). It investigates how ICL works in LLMs by analyzing contrastive demonstrations and saliency maps. The authors build contrastive demonstrations by flipping labels, perturbing input text, and adding complementary explanations. For a sentiment analysis task, they find that flipping labels is more likely to reduce salience for smaller models (e.g., GPT-2), while having an opposite impact on large models (e.g., InstructGPT). The impact of different demonstration types appears to vary depending on the model scale and task type. Further analysis is required across a range of models, tasks, and datasets. Another study investigates whether ICL in large language models is enabled by semantic priors from their pre-training or if it learns input label mappings from the provided examples (Wei et al., 2023b). Experimental results indicate that large models can override semantic priors and learn contradictory input-label mappings, while small models rely more heavily on priors. Experiments with flipped labels in ICL exemplars show that large models can learn to flip predictions, while small models cannot. These results indicate that LLMs have greater ability to learn arbitrary input-label mappings, a form of symbolic reasoning unconstrained by semantic priors, which challenges the view that ICL is solely driven by leveraging priors.

4.1.2 Explaining CoT Prompting

One study investigates how chain-of-thought (CoT) prompting affects the behavior of LLMs by analyzing the saliency scores of the input tokens (Wu et al., 2023a). Saliency scores indicate how influential each input token is on the model's output. The scores are computed using gradient-based feature attribution methods. The goal is to understand whether CoT prompting changes saliency scores compared to standard prompting, offering insights into why CoT improves performance. The analysis of saliency scores suggests that CoT prompting makes models consider question tokens in a more stable way. This more stable consideration of input may induce the generation of more consistently accurate answers compared to standard prompting. Other work has focused on perturbing CoT demonstrations in few-shot prompts, e.g., by adding errors, to determine which aspects are important for generating high-performing explanations (Madaan & Yazdanbakhsh, 2022; Wang et al., 2022a). Counterfactual prompts have been proposed to perturb key components of a prompt: symbols, patterns, and text (Madaan & Yazdanbakhsh, 2022). Experimental analysis indicates the intermediate reasoning steps act more as a beacon for the model to replicate symbols into factual answers, rather than facilitate learning to solve the task.

4.1.3 Representation Engineering

Unlike the aforementioned two lines of research that explain LLMs from the prompt engineering perspective, this family of research explains LLMs from the representation engineering perspective. Representation engineering explains models from a top-down perspective and regards representation and its transformation as the primary element of analysis. Such approaches focus on structure and the characteristic of the representation space to capture emergent representations and high-level cognitive phenomena. Zou et al. (2023) implement representation engineering in two parts: 1) representation reading, 2) representation control. Representation reading identifies representations for high-level concepts and functions within a network. Inspired by neuroimaging methodologies, linear artificial tomograph scan is adopted. To elicit concepts and functions well, prompt templates that include stimulus or instructions are designed separately. For concepts, neural activity can be collected either from representation of most representative tokes or from the last token. For functions, neural activity can be collected from the response after a certain token. Then, linear probes are introduced to predict concepts and functions with neural activity. Representation control aims to manipulate the inner representation of concepts and functions based on understanding from representation reading to meet safety requirements. Directly adding reading vectors can induce honest model output and subtracting reading vectors can induce models to lie, which demonstrates great potential in improving models. Similarly, studying the representation structures on a high-quality dataset of true/false statements also reveals the linear structure of representations. The trained probes generalize well on other datasets. As in the conclusion of the aforementioned work, the truth directions can be identified and used to induce true or false output (Marks & Tegmark, 2023). By analyzing the learned representation of six spatial or temporal datasets, LLMs such as LLaMA-13B are demonstrated to learn linear representations of space and time. In addition, similar patterns have been found in models of different sizes. The representations are also increasingly accurate as the model scales up. The model also has specialized neurons that activate as a function of space or time, which aligns with the founding of factual knowledge in LLMs (Gurnee & Tegmark, 2023). In conclusion, representation engineering could be promising techniques to control model output, but further ablation studies are still required to identify its strengths and weaknesses.

4.2 Assistant Model Explanation

Due to the large-scale unsupervised pre-training and the supervised alignment fine-tuning, LLMs belonging to this paradigm have strong reasoning ability. However, their sheer scale also makes them susceptible to generating problematic outputs such as hallucinations. Explainability research aims to 1) elucidate the role of the alignment fine-tuning, 2) analyze the causes of hallucinations, and 3) uncertainty quantification.

4.2.1 Explaining the Role of Fine-tuning

Assistant models are typically trained in two stages. First, they undergo *unsupervised pre-training* on large amounts of raw text to learn general linguistic representations. This pre-training stage allows the models to acquire general language knowledge. Second, the models go through *alignment fine-tuning* via supervised and reinforcement learning. This aligns the models with specific end tasks and user preferences. Explainability research on these models focuses on determining whether their knowledge comes predominantly from the initial pre-training stage, wherein they acquire general language abilities, or from the subsequent alignment fine-tuning stage, wherein they are tailored to specific tasks and preferences. Understanding the source of the models' knowledge provides insight into how to improve and interpret their performance.

A recent study by Zhou et al. (2023) investigated the relative importance of pre-training versus instruction fine-tuning for language models. In the experiment, the authors used only 1,000 carefully selected instructions to tune the LLaMA-65B model, without reinforcement learning, and achieved performance comparable to GPT-4. The researchers hypothesized that alignment may be a simpler process where the model learns interaction styles and formats, while almost all knowledge of LLMs is acquired during pre-training. The experimental findings demonstrated the power of pre-training and its relative importance over large-scale fine-tuning and reinforcement learning approaches. Complex fine-tuning and reinforcement learning techniques may be less crucial than previously believed. On the other hand, this study also indicates that data quality is more important compared to data quantity during instruction fine-tuning. Furthermore, Wu et al. (2023c) looked into the role of instruction fine-tuning by examining instruction following and concept-level knowledge evolution. The result shows that instruction fine-tuned models can better distinguish instruction and context, and follow users' instructions well. Besides, they can focus more on middle and tail of input prompts than pre-trained models. And fine-tuned models adjust concepts toward downstream user-oriented tasks explicitly but the linguistic distributions remain the same. Contradict to conventional belief that higher layers capture more semantic knowledge, the proportion of captured semantic knowledge initially grows then drops aggressively in fine-tuned models. In the view of self-attention heads activation, it is found that instruction fine-tuning adapts pre-trained models recognizing instruction verbs by making more neurons in the lower-level layer encode word-word patterns (Wu et al., 2023c).

Another recent study (Gudibande et al., 2023) showed that imitation can successfully improve the style, persona and ability of the language model to follow instructions, but does not improve language models on more complex dimensions such as factuality, coding, and problem solving. Imitation is another commonly used technique for training an assistant model, where a foundation model like GPT-2 or LLaMA is fine-tuned on outputs generated by a more advanced system, such as a proprietary model like ChatGPT. Furthermore, the technical report of LLaMA-2 (Touvron et al., 2023b) suggests that the fine-tuning stage mainly helps increase the helpfulness and safety of language models, where helpfulness describes how well LLaMA-2-Chat responses satisfy user requests and contain intended information, and safety refers to avoiding unsafe responses like toxic content.

Taken together, these studies emphasize the significant role of foundation models, highlighting the importance of pre-training. The findings suggest that assistant models' knowledge is mainly captured during the pre-training stage. Subsequent instruction fine-tuning then helps activate this knowledge towards useful outputs for end users. Furthermore, reinforcement learning can further align the model with human values.

4.2.2 Explaining Hallucination

The rapid development of LLMs has raised concerns about their trustworthiness, as they have the potential to exhibit undesirable behaviors such as generating hallucination, a phenomenon in which models generate output that is irrelevant and nonsensical in a natural manner (Zhang et al., 2023; Huang et al., 2023). There emerges increasing interest from the community in understanding how hallucination is produced and how to reduce hallucination generation.

Recent analysis research indicates that the hallucination phenomenon stems from various problems within datasets (Dziri et al., 2022), which can be categorized into two main classes: 1) a lack of relevant data, 2) repeated data. For example, long-tail knowledge is prevalent in training data and LLMs easily fall short in learning such knowledge (Kandpal et al., 2023). On the other hand, deduplicating data is challenging to be done perfectly. Duplicate data within the training dataset can noticeably impair the model's performance. Hernandez et al. (2022) find that the performance of an 800M parameter model can degrade to that of a 400M parameter model by only repeating 10% of the training data. When examining the model's performance in terms of scaling laws, a certain range of repetition frequency in the middle could have a detrimental impact. This range is hypothesized to lead the model to memorize the data and consequently consume a large portion of its capacity.

Moreover, recent studies find that hallucination also arises from certain limitations inherent to models. McKenna et al. (2023) demonstrate that LLMs still rely on memorization at the sentence level and statistical patterns at the corpora level instead of robust reasoning. This is evidenced by their analysis of various LLM families' performance on natural language inference tasks. Further, Wu et al. (2023b) reveal that LLMs are imperfect in both memorization and reasoning regarding ontological knowledge. Berglund et al. (2023) points out that LLMs usually suffer from logical deduction due to reversal curse. LLMs tend to be overconfident in their output and struggle to identify the factual knowledge boundary precisely (Ren et al., 2023). Furthermore, LLM favors co-occurrence words over factual answers, a phenomenon often referred to as shortcuts or spurious correlations (Kang & Choi, 2023). Similarly, another undesired behavior sycophancy also exists in LLMs, which refers to models could generate answers aligning users' view rather than facts. The worst thing is that model scaling and instruction tuning could increase such behavior (Wei et al., 2023a).

There are several ways to address the hallucination problem. Firstly, scaling is always a good step to take. The performance of PaLM with 540 billion parameters steeply increased on a variety of tasks. Even it also suffers from learning long-tail knowledge, but its memorization abilities are shown to be better than small models (Chowdhery et al., 2022). In text summarization tasks, Ladhak et al. (2023) shows that using more extractive fine-tuning datasets and adapter-fine-tuning that fine-tunes part of parameters usually generates less hallucinations but will not change the distribution of hallucination. Consequently, mitigation can be implemented by either data side such as improve fine-tuning datasets and add synthetic-data intervention (Wei et al., 2023a), or on the model side, such as different optimization approaches.

4.2.3 Uncertainty Quantification

There is also growing interest within the research community in quantifying the uncertainty of LLM predictions, to better understand the reliability and limitations of these powerful models.

Most existing literature on uncertainty quantification focuses on logits, which is however less suitable for LLMs, especially closed-source ones. This necessitates non-logit-based approaches for eliciting uncertainty in LLMs, known as confidence elicitation (Xiong et al., 2023). There are several representative methods for uncertainty estimation of LLMs. First, consistency-based uncertainty estimation involves generating multiple responses to a question and using the consistency between these responses to estimate the model's confidence (Xiong et al., 2023). Specifically, it introduces randomness into the answer generation process (self-consistency) or adds misleading hints to the prompt (induced-consistency) to produce varied responses. The more consistent the multiple responses are, the higher the estimated confidence in the answer is. Second, LLMs can deliver their confidence verbally by providing direct and specific responses to indicate high confidence in their predictions, and giving indirect, vague, or ambiguous responses to convey lower confidence. LLMs can explicitly state a percentage to quantify their confidence level. For example, "I am only 20% confident in this answer" clearly communicates low confidence (Xiong et al., 2023). Third, the uncertainty can be aggregated from token-level uncertainty (Duan et al., 2023). LLMs generate text by predicting each token, which can be framed as a classification task. Token-level uncertainty methods calculate a confidence score for each predicted token based on its probability distribution. Then the overall uncertainty can be estimated based on the aggregation of token-level uncertainties.

4.3 Making Use of Explanations

In this section, we discuss techniques to harness the explanatory abilities of prompting-based LLMs to improve the predictive performance of language models and enable beneficial applications.

4.3.1 Improving LLMs

This line of research investigates whether LLMs can benefit from explanations when learning new tasks from limited examples. Specifically, it investigates whether providing explanations for the answers to few-shot tasks can improve the model's performance on these tasks (Lampinen et al., 2022). Two forms of explanations are provided: *pre-answer explanations* and *post-answer explanations*. Wei et al. (2022) proposes a method called chain-of-thought prompting, which provides intermediate reasoning steps as explanations in prompts before the answers. This has helped language models achieve state-of-the-art results in arithmetic, symbolic, and common-sense reasoning tasks. Another recent study provides explanations after the answer in the prompts (Lampinen et al., 2022). Experimental analysis indicates that providing explanations can improve the few-shot learning performance of large language models, but the benefits depend on model scale and explanation quality. Additionally, customizing explanations specifically for the task using the validation set further increases their benefits (Lampinen et al., 2022).

Another recent study proposes explanation tuning, an approach that trains smaller language models using detailed step-by-step explanations from more advanced models as a form of supervision (Mukherjee et al., 2023). Section 4.2.1 indicates that imitation tuning mainly allows smaller models to learn the style of the larger models rather than the reasoning process. To address this limitation, this work proposes leveraging richer signals beyond just input-output pairs to teach smaller models to mimic the reasoning process of large foundation models like GPT-4. Specifically, the authors collect training data consisting of prompts and detailed explanatory responses from GPT-4. To allow GPT-4 to generate explanations, system instructions such as "`You are a helpful assistant, who always provides explanation. Think like you are answering to a five-year-old.`" are utilized. Experimental results indicate that models trained with explanation tuning outperform models trained using conventional instruction tuning in complex zero-shot reasoning benchmarks like BigBench Hard.

The insights captured from the explanations can also be utilized to compress the instructions (Yin et al., 2023). The authors use ablation analysis to study the contribution of different categories of content in task definitions. The insights from the ablation analysis can then be utilized to compress the task instruction. Take classification task as an example, the analysis indicates that the most important components within the task instruction are label-relevant information. Removing other contents will only slightly impact the classification performance, and the authors find that model performance only drops substantially when removing output label information. Additionally, they propose an algorithm to automatically compress definitions by removing unnecessary tokens, finding 60% can be removed while maintaining or improving performance for T5-XL model on a holdout dataset.

Moreover, some studies have also delved into the effectiveness of the explanations generated by LLMs in enhancing few-shot in-context learning. For multi-step symbolic reasoning tasks, involving code execution and arithmetic operations, Nye et al. (2021) found that incorporating intermediate computation steps can significantly boost the model's ability. On the flip side, when it comes to textual reasoning tasks including question answering and natural language inference, only text-davinci-002 was observed an increase in accuracy. The other four models, including OPT, GPT-3(davinci), InstructGPT(text-davinci-001), and text-davinci-002, did not show a clear improvement and even performed worse. The explanations generated by LLMs are assessed in two dimensions: factuality and consistency. The result reveals that LLMs can generate unrealistic explanations but still align with predictions, which in turn leads to incorrect predictions (Ye & Durrett, 2022). Building on top of the finding, an explanation optimization framework has been proposed to select explanations that lead to high performance (Ye & Durrett, 2023). Therefore, improving the accuracy of model predictions requires the generation of reliable explanations by LLMs, which remains a great challenge at this time.

4.3.2 Downstream Applications

Explainability can also be applied to real-world problems such as education, finance, and healthcare. For example, explainable zero-shot medical diagnosis is an interesting use case. One recent study proposes a framework for explainable zero-shot medical image classification utilizing vision-language models like CLIP along with LLMs like ChatGPT (Liu et al., 2023a). The key idea is to leverage ChatGPT to automatically generate detailed textual descriptions of disease symptoms and visual features beyond just the disease name. This additional textual information helps to provide more accurate and explainable diagnoses from CLIP (Radford et al., 2021). To handle potential inaccuracies from ChatGPT on medical topics, the authors design prompts to obtain high-quality textual descriptions of visually identifiable symptoms for each disease class. Extensive experiments on multiple medical image datasets demonstrate the effectiveness and explainability of this training-free diagnostic pipeline.

5 Explanation Evaluation

In previous sections, we introduced different explanation techniques and their usages, but evaluating how faithfully they reflect a model's reasoning process remains a challenge. We roughly group the evaluations into two families: evaluation of local explanation for traditional fine-tuning paradigm (Section 5.1) and evaluation of natural language CoT explanations for prompting paradigm (Section 5.2). Two key dimensions of evaluations are plausibility to humans and faithfulness in capturing LLMs' internal logic.

Technically, evaluating explanation involves human or automated model approaches. Human evaluations assess plausibility through similarity between model rationales and human rationales or subjective judgments. However, these methods usually neglect faithfulness. Subjective judgments may also not align with model reasoning, making such an evaluation unreliable. As argued by Jacovi & Goldberg (2020), faithful evaluation should have a clear goal and avoid human involvement. Automatic evaluations test importance by perturbing model rationales, avoiding human biases. Therefore, developing rigorous automatic metrics is critical for fair faithfulness evaluation, which will be covered under the faithfulness evaluation dimension.

5.1 Explanation Evaluations in Traditional Fine-tuning Paradigms

We introduce the evaluation of the local explanation from two aspects: plausibility and faithfulness. Both parts will mainly cover universal properties and metrics that can be applied to compare various explanation approaches. We focus on quantitative evaluation properties and metrics, which are usually more reliable than qualitative evaluations.

Evaluating plausibility The plausibility of local explanation is typically measured at the input text or token level. Plausibility evaluation can be categorized into five dimensions: grammar, semantics, knowledge, reasoning, and computation (Shen et al., 2022). These dimensions describe the relationship between the masked input and human-annotated rationales. Different evaluation dimensions require different kinds of datasets. For example, a sentence "`The country [MASK] was established on July 4, 1776.`" has the human-annotated rationale "`established on July 4, 1776`" and the answer to the mask should be "`the United States`" deriving from fact/knowledge. Although rationales might be in different granularity levels such as token or snippet and dimensions, evaluation procedures are the same except for diversified metrics. Human-annotated rationales are generally from benchmark datasets, which should meet several criteria: 1) sufficiency meaning rationales are enough for people to make correct prediction; 2) compactness requiring that if any part in the rationales is removed, the prediction will change (Mathew et al., 2021). The explanation models are then responsible for predicting important tokens and generating rationales with these tokens. The above two kinds of rationales will be measured with various metrics. Popular metrics can be classified into two classes according to their scope of measurement. Metrics measuring two token-level rationales include Intersection-Over-Union (IOU), precision and recall. Metrics that measure overall plausibility include the F1 score for discrete cases and the area under the precision recall curve (AUPRC) for continuous or soft token selection cases (DeYoung et al., 2020).

Evaluating Faithfulness Evaluation principles and metrics provide a unified way to measure faithfulness quantitatively. Since they are often defined for specific explanation techniques, we will cover only some common yet universal principles from the model perspective and metrics from the data perspective.

There are several model-level principles to which explanation methods should adhere in order to be faithful, which include implementation invariance, input invariance, input sensitivity, completeness, polarity consistency, prediction consistency and sufficiency. Implementation invariance also known as model sensitivity means that the attribution scores should remain the same regardless of differences in the model architectures, as long as the networks are functionally equal (Sundararajan et al., 2017). Even gradient-based approaches usually meet this metric well; the assumption may not be grounded. Input invariance requires attribution methods to reflect sensitivity of prediction models w.r.t. effective input changes. For example, attribution scores should remain the same over constant shift of the input (Kindermans et al., 2017). Input sensitivity defines attribution scores should be non-zero for features that solely explain prediction differences (Sundararajan et al., 2017). Completeness combines sensitivity and implementation invariance with path integrals from calculus (Sundararajan et al., 2017), which only apply to differentiable approaches. Polarity consistency points out that some high-ranking features could impose suppression effects on final predictions, which negatively impacts explanations and should be avoided, but mostly not (Liu et al., 2022). Prediction consistency confines that instances with same explanations should have the same prediction. And sufficiency requires that data with same attributions should have same related labels even with different explanations (Dasgupta et al., 2022). In this class of approaches, researchers aim at preventing certain types of contradictory explanations by formulating axioms and properties. However, each metric can address only one particular facet of faithfulness problems. It is extremely difficult to provide all-in-one solutions within a single framework. Additionally, these approaches focus solely on avoiding inconsistent behaviors of explanation models by designing properties for explanation methods. The overall performance of models is measured with the following metrics.

A prominent line of model-agnostic work measures faithfulness by quantitatively verifying the relationship between prediction and model rationales. Some common metrics calculated on the test set are as follows:

- Comprehensiveness (COMP): the change in probability for the original predicted class before and after top-ranked important tokens removed, which means how influential the rationale is. It is formulated as comprehensiveness $= m\left(x_i\right)_j - m\left(x_i \backslash r_i\right)_j$. A higher score indicates the importance of rationales/tokens (DeYoung et al., 2020).
- Sufficiency (SUFF): the degree to which the parts within the extracted rationales can allow the model to make a prediction, which is defined as sufficiency $= m\left(x_i\right)_j - m\left(r_i\right)_j$ (DeYoung et al., 2020).
- Decision Flip - Fraction Of Tokens (DFFOT): the average fraction of tokens removed to trigger a decision flip (Chrysostomou & Aletras, 2021).
- Decision Flip - Most Informative Token (DFMIT): the rate of decision flips caused by removing the most influential token (Chrysostomou & Aletras, 2021).

In ERASER (DeYoung et al., 2020), related tokens are classified into groups ranked by importance scores so that tokens can be masked in a ranked order and gradually observe output changes. The correlation between output changes and the importance of masked tokens denotes models' ability in correctly attributing feature importance. As claimed by TaSc (Chrysostomou & Aletras, 2021), higher DFMIT and lower DFFOT are preferred, as important tokens are precisely identified and models are more faithful. In contrast, some work measures faithfulness through weaknesses in explanations such as shortcut learning and polarity of feature importance. Bastings et al. (2022) quantifies the faithfulness by how well the model identifies learned shortcuts. In this case, metrics like *precision@k* (the percentage of shortcuts in top-k tokens) and *mean rank* (the average depth searched in salience ranking) signifies how well the top features represent all ground truth shortcuts. Likewise, higher precison@k and lower mean rank indicate good faithfulness of the models. Liu et al. (2022) examine faithfulness by performing the violation test to make sure the model correctly reflects feature importance and feature polarity.

There are two key questions that persist when evaluating explanation models, regardless of the specific metrics used: 1) how well does the model quantify important features? 2) can the model effectively and correctly extract as many influential features as possible from the top-ranked features? However, existing evaluation metrics are often inconsistent with the same explanation models. For example, the best-ranked explanation by DFFOT could be the worst with SUFF (Chan et al., 2022b). TaSc demonstrates that attention-based importance metrics are more robust than non-attention ones whereas regarding attention as an explanation is still debatable (Jain & Wallace, 2019).

Additionally, these evaluation metrics cannot be applied directly to natural language explanations, as such explanations rarely have a straightforward relationship to the inputs. Atanasova et al. (2023) proposes two faithfulness tests for natural language explanation models. One test is the counterfactual test, where counterfactual examples are constructed from the original example by inserting tokens that change the prediction. If words from inserted tokens are not present in the explanation, the explanation approach is deemed unfaithful. Another test is the input reconstruction test, which explores whether the explanation is sufficient to make the same prediction as the original example. The explanation for each example is transformed into a new input given the original input and the explanation itself. Unfortunately, because both tests can introduce new linguistic variants, they struggle with evaluating faithfulness fairly when new phrases are generated. Alternatively, Rev (Chen et al., 2023b) provides evaluation metrics from the perspective of information by examining whether natural language explanations support model predictions and whether new information from explanations justify model predictions.

5.2 Evaluation of Explanations in Prompting Paradigms

Recently, LLMs such as GPT-3 and GPT-4 have shown impressive abilities to generate natural language explanations for their predictions. However, it remains unclear whether these explanations actually help humans understand the model's reasoning process and generalize to new inputs. Note that the goals and perspectives of evaluating such explanations (e.g., CoT rationales) are different from those of evaluating traditional explanations introduced in Section 5.1 (Golovneva et al., 2022; Prasad et al., 2023). Metrics such as plausibility, faithfulness and stability also known as diversity have been developed to evaluate explanation. Similar to traditional explanations, we focus on evaluating plausibility and faithfulness.

Evaluating Plausibility One recent work studies whether explanations satisfy human expectations and proposes to evaluate the counterfactual simulatability of natural language explanations (Chen et al., 2023d). That is, whether an explanation helps humans infer how an AI model will behave on diverse counterfactual inputs. They implement two metrics: simulation generality (diversity of counterfactuals the explanation helps simulate) and simulation precision (fraction of simulated counterfactuals where human guess matches model output). They find that explanations from LLMs such as GPT-3.5 and GPT-4 have low precision, indicating that they mislead humans to form incorrect mental models. The paper reveals limitations of current methods and that optimizing human preferences like plausibility may be insufficient for improving counterfactual simulatability.

Evaluating Faithfulness This line of studies the faithfulness of explanations, i.e., examining how well explanations reflect the actual reasons behind a model's predictions. For example, experimental analysis of one recent study indicates that the chain-of-thought explanation can be systematically unfaithful (Turpin et al., 2023). The authors introduced bias into model inputs by reordering multiple choice options in few-shot prompts to make the answer always "(A)". However, language models like GPT-3.5 and Claude 1.0 failed to acknowledge the influence of these biased features in their explanations. The models generated explanations that did not faithfully represent the true decision-making process. Another work also indicates that the LLM's stated CoT reasoning could be unfaithful on some tasks, and smaller models tend to generate more faithful explanations compared to larger and more capable models (Lanham et al., 2023). These research highlights concerns about the faithfulness of explanations from LLMs, even when they appear sensible. To improve reasoning faithfulness over CoT, one preliminary study proposes to generate models reasoning by decomposing questions into subquestions and answering them separately (Radhakrishnan et al., 2023). The analysis indicates that decomposition methods can approach CoT's performance while increasing faithfulness on several metrics. More future research is needed to develop methods to make model explanations better reflect the underlying reasons for predictions.

6 Research Challenges

In this section, we explore key research challenges that warrant further investigation from both the NLP and the explainable AI communities.

6.1 Explanation without Ground Truths

Ground truth explanations for LLMs are usually inaccessible. For example, there are currently no benchmark datasets to evaluate the global explanation of individual components captured by LLMs. This presents two main challenges. First, it is difficult to design explanation algorithms that accurately reflect an LLM's decision-making process. Second, the lack of ground truth makes evaluating explanation faithfulness and fidelity problematic. It is also challenging to select a suitable explanation among various methods in the absence of ground truth guidance. Potential solutions include involving human evaluations and creating synthetic explanatory datasets.

6.2 Sources of Emergent Abilities

LLMs exhibit surprising new capabilities as the model scale and training data increases, even without being explicitly trained to perform these tasks. Elucidating the origins of these emergent abilities remains an open research challenge, especially for proprietary models like ChatGPT and Claude whose architectures and training data are unpublished. Even open-source LLMs like LLaMA currently have limited interpretability into the source of their emergent skills. This can be investigated from both a model and a data perspective.

Model Perspective It is crucial to further investigate the Transformer-based model to shed light on the inner workings of LLMs. Key open questions include: 1) What specific model architectures give rise to the impressive emergent abilities of LLMs? 2) What is the minimum model complexity and scale needed to achieve strong performance across diverse language tasks? Continuing to rigorously analyze and experiment with foundation models remains imperative as LLMs continue to rapidly increase in scale. Advancing knowledge in these areas will enable more controllable and reliable LLMs. This can provide hints as to whether there will be new emergent abilities in the near future.

Data Perspective In addition to the model architecture, training data is another important perspective for understanding the emergent abilities of LLMs. Some representative research questions include: 1) Which specific subsets of the massive training data are responsible for particular model predictions, and is it possible to locate these examples? 2) Are emergent abilities the result of model training or an artifact of data contamination issues (Blevins et al., 2023)? 3) Are training data quality or quantity more important for effective pre-training and fine-tuning of LLMs? Understanding the interplay between training data characteristics and the resulting behavior of the model will provide key insights into the source of emergent abilities in large language models.

6.3 Comparing Two Paradigms

For a given task such as natural language inference (NLI), the downstream fine-tuning paradigm and prompting paradigm can demonstrate markedly different in-distribution and out-of-distribution (OOD) performance. This suggests that the two approaches rely on divergent reasoning for predictions. However, a comprehensive comparison of explanations between fine-tuning and prompting remains lacking. Further research is needed to better elucidate the explanatory differences between these paradigms. Some interesting open questions include: 1) How do fine-tuned models and prompted models differ in the rationales used for prediction on in-distribution examples? and 2) What causes the divergence in OOD robustness between fine-tuning and prompting? Can we trace this back to differences in reasoning? Advancing this understanding will enable selecting the right paradigm for given use cases and improving robustness across paradigms.

6.4 Shortcut Learning of LLMs

Recent explainability research indicates that language models often take shortcuts when making predictions. For the downstream fine-tuning paradigm, studies show that language models leverage various dataset artifacts and biases for natural language inference tasks, such as lexical bias, overlap bias, position bias, and style bias (Du et al., 2023). This significantly impacts out-of-distribution generalization performance. For the prompting paradigm, a recent study analyzes how language models use longer contexts (Liu et al., 2023b). The results show that performance was highest when relevant information was at the beginning or end of the context, and worsened when models had to access relevant information in the middle of long contexts. These analyses demonstrate that both paradigms tend to exploit shortcuts in certain scenarios, highlighting the need for more research to address this problem and improve generalization capabilities.

6.5 Attention Redundancy

Recent research has investigated attention redundancy using interpretability techniques in large language models for both traditional fine-tuning and prompting paradigms (Bian et al., 2021; Bansal et al., 2022). For example, Bian et al. analyze attention redundancy across different pretraining and fine-tuning phases using BERT-base (Bian et al., 2021). Experimental analysis indicates that there is attention redundancy, finding that many attention heads are redundant and could be pruned with little impact on downstream task performance. Similarly, Bansal et al. investigate attention redundancy in terms of the in-context learning scenario using OPT-66B (Bansal et al., 2022). They found that there is redundancy in both attention heads and feedforward networks. Their findings suggest that many attention heads and other components are redundant. This presents opportunities to develop model compression techniques that prune redundant modules while preserving performance on downstream tasks.

6.6 Shifting from Snapshot Explainability to Temporal Analysis

There is also an viewpoint that current interpretability research neglect the training dynamics. Existing research is mainly post-hoc explanation on fully trained models. The lack of developmental investigation on training process can generate biased explanation by failing in targeting emerging abilities or vestigial parts that convergence counts on, namely phase transitions. Besides, performing interventions on certain features fail to reflect interactions between features (Saphra, 2022). Therefore, there is a trend shifting from static, snapshot explainability analysis to dynamic, temporal analysis. By examining several checkpoints during training, Chen et al. (2023a) identified an abrupt pre-training window wherein models gain Syntactic Attention Structure (SAS), which occurs when a specialized attention head focus on a word's syntactic neighbors, and meanwhile a steep drop in training loss. They also showed that SAS is critical for acquiring grammatical abilities during learning. Inspired by such a perspective, development analysis could uncover more casual relations and training patterns in the training process that are helpful in understanding and improving model performance.

6.7 Safety and Ethics

The lack of interpretability in LLMs poses significant ethical risks as they become more capable. Without explainability, it becomes challenging to analyze or constrain potential harms from issues such as misinformation, bias, and social manipulation. Explainable AI techniques are vital to audit these powerful models and ensure alignment with human values. For example, tools to trace training data attribution or visualize attention patterns can reveal embedded biases, such as gender stereotypes (Li et al., 2023a). Additionally, probing classifiers can identify if problematic associations are encoded within the model's learned representations. Researchers, companies, and governments deploying LLMs have an ethical responsibility to prioritize explainable AI. Initiatives such as rigorous model audits, external oversight committees, and transparency regulations can help mitigate risks as LLMs become more prevalent. For example, as alignment systems continue to grow in scale, human feedback is becoming powerless at governing them, posing tremendous challenges for the safety of these systems. As claimed by Martin (2023), leveraging explainability tools as part of audit processes to supplement human feedback could be a productive approach. Advancing interpretability techniques must remain a priority alongside expanding model scale and performance to ensure the safe and ethical development of increasingly capable LLMs.

7 Conclusions

In this paper, we have presented a comprehensive overview of explainability techniques for LLMs. We summarize methods for local and global explanations based on model training paradigms. We also discuss using explanations to improve models, evaluation, and key challenges. Major future development options include developing explanation methods tailored to different LLMs, evaluating explanation faithfulness, and improving human interpretability. As LLMs continue to advance, explainability will become incredibly vital to ensure these models are transparent, fair, and beneficial. We hope that this survey provides a useful organization of this emerging research area, as well as highlights open problems for future work.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, Guilherme Penedo, et al. Falcon-40b: an open large language model with state-of-the-art performance. 2023.

Anthropic. Decomposing language models into understandable components? https://www.anthropic.com/index/decomposing-language-models-into-understandable-components, 2023.

AnthropicAI. Introducing claude, 2023. URL https://www.anthropic.com/index/introducing-claude.

Omer Antverg and Yonatan Belinkov. On the Pitfalls of Analyzing Individual Neurons in Language Models, August 2022. URL http://arxiv.org/abs/2110.07483. arXiv:2110.07483 [cs].

Marianna Apidianaki and Aina Garí Soler. ALL Dolphins Are Intelligent and SOME Are Friendly: Probing BERT for Nouns' Semantic Properties and their Prototypicality, October 2021. URL http://arxiv.org/abs/2110.06376. arXiv:2110.06376 [cs].

Ines Arous, Ljiljana Dolamic, Jie Yang, Akansha Bhardwaj, Giuseppe Cuccu, and Philippe Cudré-Mauroux. Marta: Leveraging human rationales for explainable text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 5868–5876, 2021.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. Faithfulness Tests for Natural Language Explanations, June 2023. URL http://arxiv.org/abs/2305.18029. arXiv:2305.18029 [cs].

Bing Bai, Jian Liang, Guanhua Zhang, Hao Li, Kun Bai, and Fei Wang. Why attentions may not be interpretable? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 25–34, 2021.

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. *arXiv preprint arXiv:2212.09095*, 2022.

Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. Grad-SAM: Explaining Transformers via Gradient Self-Attention Maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2882–2887, Virtual Event Queensland Australia, October 2021. ACM. ISBN 978-1-4503-8446-9. doi: 10.1145/3459637.3482126. URL https://dl.acm.org/doi/10.1145/3459637.3482126.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. "Will You Find These Shortcuts?" A Protocol for Evaluating the Faithfulness of Input Salience Methods for Text Classification, November 2022. URL http://arxiv.org/abs/2111.07367. arXiv:2111.07367 [cs].

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Identifying and Controlling Important Neurons in Neural Machine Translation, November 2018. URL http://arxiv.org/abs/1811.01157. arXiv:1811.01157 [cs].

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, April 2022. ISSN 0891-2017, 1530-9312. doi: 10.1162/coli_a_00422. URL https://direct.mit.edu/coli/article/48/1/207/107571/Probing-Classifiers-Promises-Shortcomings-and.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 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 (Volume 1: Long Papers)*, pp. 1–10, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-1001.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A", September 2023. URL http://arxiv.org/abs/2309.12288. arXiv:2309.12288 [cs].

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. On attention redundancy: A comprehensive study. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2021.

Terra Blevins, Omer Levy, and Luke Zettlemoyer. Deep RNNs Encode Soft Hierarchical Syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 14–19, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2003. URL https://aclanthology.org/P18-2003.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2023. URL https://arxiv.org/abs/2211.07830.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. https://transformer-circuits.pub/2023/monosemantic-features/index.html, 2023.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. *arXiv preprint arXiv:1908.04211*, 2019.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Captum. Testing with concept activation vectors (tcav) on sensitivity classification examples and a convnet model trained on imdb dataset. https://github.com/pytorch/captum/blob/master/tutorials/TCAV_NLP.ipynb.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Anas Awadalla, Pang Wei Koh, Daphne Ippolito, Katherine Lee, Florian Tramer, et al. Are aligned neural networks adversarially aligned? *arXiv preprint arXiv:2306.15447*, 2023.

Aaron Chan, Maziar Sanjabi, Lambert Mathias, Liang Tan, Shaoliang Nie, Xiaochang Peng, Xiang Ren, and Hamed Firooz. Unirex: A unified learning framework for language model rationale extraction. In *International Conference on Machine Learning*, pp. 2867–2889. PMLR, 2022a.

Chun Sik Chan, Huanqi Kong, and Liang Guanqing. A comparative study of faithfulness metrics for model interpretability methods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5029–5038, 2022b.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer Interpretability Beyond Attention Visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 782–791, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.00084. URL https://ieeexplore.ieee.org/document/9577970/.

Angelica Chen, Ravid Schwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. *arXiv preprint arXiv:2309.07311*, 2023a.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. Probing BERT in Hyperbolic Spaces, April 2021. URL http://arxiv.org/abs/2104.03869. arXiv:2104.03869 [cs].

Hanjie Chen and Yangfeng Ji. Adversarial training for improving model robustness? look at both prediction and interpretation. In *The 36th AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. Rev: information-theoretic evaluation of free-text rationales. *The 61th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023b.

Hugh Chen, Ian C. Covert, Scott M. Lundberg, and Su-In Lee. Algorithms to estimate Shapley value feature attributions. *Nature Machine Intelligence*, 5(6):590–601, May 2023c. ISSN 2522-5839. doi: 10.1038/s42256-023-00657-x. URL https://www.nature.com/articles/s42256-023-00657-x.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. Do models explain themselves? counterfactual simulatability of natural language explanations, 2023d.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways, October 2022. URL http://arxiv.org/abs/2204.02311. arXiv:2204.02311 [cs].

George Chrysostomou and Nikolaos Aletras. Improving the Faithfulness of Attention-based Explanations with Task-specific Information for Text Classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 477–488, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.40. URL https://aclanthology.org/2021.acl-long.40.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *International Conference on Learning Representations (ICLR)*, 2020.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6309–6317, July 2019. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v33i01.33016309. URL https://ojs.aaai.org/index.php/AAAI/article/view/4592.

Sanjoy Dasgupta, Nave Frost, and Michal Moshkovitz. Framework for Evaluating Faithfulness of Local Explanations. 2022.

Joseph F. DeRose, Jiayao Wang, and Matthew Berger. Attention Flows: Analyzing and Comparing Attention Mechanisms in Language Models, September 2020. URL http://arxiv.org/abs/2009.07053. arXiv:2009.07053 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019a.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019b.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Timothy Dozat and Christopher D. Manning. Deep Biaffine Attention for Neural Dependency Parsing, March 2017. URL http://arxiv.org/abs/1611.01734. arXiv:1611.01734 [cs].

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019a.

Mengnan Du, Ninghao Liu, Fan Yang, Shuiwang Ji, and Xia Hu. On Attribution of Recurrent Neural Network Predictions via Additive Decomposition, March 2019b. URL http://arxiv.org/abs/1903.11245. arXiv:1903.11245 [cs].

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. Towards interpreting and mitigating shortcut learning behavior of nlu models. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding. *Communications of the ACM (CACM)*, 2023.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*, 2023.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models? In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5271–5285, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.387. URL https://aclanthology.org/2022.naacl-main.387.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A Mathematical Framework for Transformer Circuits — transformer-circuits.pub. https://transformer-circuits.pub/2021/framework/index.html, December 2021. [Accessed 27-11-2023].

Joseph Enguehard. Sequential Integrated Gradients: a simple but effective method for explaining language models, May 2023. URL http://arxiv.org/abs/2305.15853. arXiv:2305.15853 [cs].

Kawin Ethayarajh and Dan Jurafsky. Attention Flows are Shapley Value Explanations, May 2021. URL http://arxiv.org/abs/2105.14652. arXiv:2105.14652 [cs].

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of Neural Models Make Interpretations Difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3719–3728, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1407. URL https://aclanthology.org/D18-1407.

Siddhant Garg and Goutham Ramakrishnan. BAE: BERT-based Adversarial Examples for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6174–6181, 2020. doi: 10.18653/v1/2020.emnlp-main.498. URL http://arxiv.org/abs/2004.01970. arXiv:2004.01970 [cs].

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*, 2022.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*, 2022.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*, 2020.

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

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-Attention Attribution: Interpreting Information Interactions Inside Transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12963–12971, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i14.17533. URL https://ojs.aaai.org/index.php/AAAI/article/view/17533. Number: 14.

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with sgd. *Advances in Neural Information Processing Systems*, 32, 2019.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling Laws and Interpretability of Learning from Repeated Data, May 2022. URL http://arxiv.org/abs/2205.10487. arXiv:2205.10487 [cs].

John Hewitt and Percy Liang. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

John Hewitt and Christopher D. Manning. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 187–196, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.22. URL https://aclanthology.org/2020.acl-demos.22.

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*, 2023.

Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL https://aclanthology.org/2020.acl-main.386.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation, May 2019. URL http://arxiv.org/abs/1902.10186. arXiv:1902.10186 [cs].

Theo Jaunet, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf. VisQA: X-raying Vision and Language Reasoning in Transformers, July 2021. URL http://arxiv.org/abs/2104.00926. arXiv:2104.00926 [cs].

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://aclanthology.org/P19-1356.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Brihi Joshi, Aaron Chan, Ziyi Liu, Shaoliang Nie, Maziar Sanjabi, Hamed Firooz, and Xiang Ren. Er-test: Evaluating explanation regularization methods for language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3315–3336, 2022.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large Language Models Struggle to Learn Long-Tail Knowledge, July 2023. URL http://arxiv.org/abs/2211.08411. arXiv:2211.08411 [cs].

Cheongwoong Kang and Jaesik Choi. Impact of Co-occurrence on Factual Knowledge of Large Language Models, October 2023. URL http://arxiv.org/abs/2310.08256. arXiv:2310.08256 [cs] version: 1.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the Difference that Makes a Difference with Counterfactually-Augmented Data. 2020.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV), June 2018. URL http://arxiv.org/abs/1711.11279. arXiv:1711.11279 [stat].

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (Un)reliability of saliency methods, November 2017. URL http://arxiv.org/abs/1711.00867. arXiv:1711.00867 [cs, stat].

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Enja Kokalj, Blaž Škrlj, Nada Lavrač, Senja Pollak, and Marko Robnik-Šikonja. BERT meets Shapley: Extending SHAP Explanations to Transformer-based Classifiers. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pp. 16–21, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.hackashop-1.3.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the Dark Secrets of BERT, September 2019. URL http://arxiv.org/abs/1908.08593. arXiv:1908.08593 [cs, stat].

Nicholas Kroeger, Dan Ley, Satyapriya Krishna, Chirag Agarwal, and Himabindu Lakkaraju. Are large language models post hoc explainers? *arXiv preprint arXiv:2310.05797*, 2023.

Jenny Kunz and Marco Kuhlmann. Classifier Probes May Just Learn from Linear Context Features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5136–5146, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.450. URL https://aclanthology.org/2020.coling-main.450.

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen McKeown, and Tatsunori Hashimoto. When Do Pre-Training Biases Propagate to Downstream Tasks? A Case Study in Text Summarization. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3206–3219, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.234. URL https://aclanthology.org/2023.eacl-main.234.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 537–563, 2022.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Dong-Ho Lee, Akshen Kadakia, Brihi Joshi, Aaron Chan, Ziyi Liu, Kiran Narahari, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. Xmd: An end-to-end framework for interactive explanation-based debugging of nlp models, 2022.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. How is BERT surprised? Layerwise detection of linguistic anomalies, May 2021a. URL http://arxiv.org/abs/2105.07452. arXiv:2105.07452 [cs].

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. Contextualized Perturbation for Textual Adversarial Attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5053–5069, Online, 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.400. URL https://aclanthology.org/2021.naacl-main.400.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Probing via Prompting, July 2022. URL http://arxiv.org/abs/2207.01736. arXiv:2207.01736 [cs].

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 681–691, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1082. URL https://aclanthology.org/N16-1082.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding Neural Networks through Representation Erasure, January 2017. URL http://arxiv.org/abs/1612.08220. arXiv:1612.08220 [cs].

Yingji Li, Mengnan Du, Rui Song, Xin Wang, and Ying Wang. A survey on fairness in large language models. *arXiv preprint arXiv:2308.10149*, 2023a.

Zongxia Li, Paiheng Xu, Fuxiao Liu, and Hyemi Song. Towards understanding in-context learning with contrastive demonstrations and saliency maps, 2023b.

Tom Lieberum, Matthew Rahtz, János Kramár, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: getting inside bert's linguistic knowledge. *arXiv preprint arXiv:1906.01698*, 2019.

Jiaxiang Liu, Tianxiang Hu, Yan Zhang, Xiaotang Gai, Yang Feng, and Zuozhu Liu. A chatgpt aided explainable framework for zero-shot medical image diagnosis, 2023a.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023b.

Ninghao Liu, Yunsong Meng, Xia Hu, Tie Wang, and Bo Long. Are interpretations fairly evaluated? a definition driven pipeline for post-hoc interpretability. *arXiv preprint arXiv:2009.07494*, 2020.

Yibing Liu, Haoliang Li, Yangyang Guo, Chenqi Kong, Jing Li, and Shiqi Wang. Rethinking Attention-Model Explainability through Faithfulness Violation Test. 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017a.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017b. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Daniel D. Lundstrom, Tianjian Huang, and Meisam Razaviyayn. A Rigorous Study of Integrated Gradients Method and Extensions to Internal Neuron Attributions. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 14485–14508. PMLR, June 2022. URL https://proceedings.mlr.press/v162/lundstrom22a.html. ISSN: 2640-3498.

Siwen Luo, Hamish Ivison, Caren Han, and Josiah Poon. Local Interpretations for Explainable Natural Language Processing: A Survey, October 2022. URL http://arxiv.org/abs/2103.11072. arXiv:2103.11072 [cs].

Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, Himabindu Lakkaraju, et al. Post hoc explanations of language models can improve language models. *arXiv preprint arXiv:2305.11426*, 2023.

Aman Madaan and Amir Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*, 2022.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Sammy Martin. Ten Levels of AI Alignment Difficulty, 2023. URL https://www.lesswrong.com/posts/EjgfreeibTXRx9Ham/ten-levels-of-ai-alignment-difficulty. [Accessed 21-08-2023].

Rebecca Marvin and Tal Linzen. Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1151. URL https://aclanthology.org/D18-1151.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 14867–14875, 2021.

Rowan Hall Maudslay and Ryan Cotterell. Do Syntactic Probes Probe Syntax? Experiments with Jabberwocky Probing, June 2021. URL http://arxiv.org/abs/2106.02559. arXiv:2106.02559 [cs].

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. Sources of Hallucination by Large Language Models on Inference Tasks, October 2023. URL http://arxiv.org/abs/2305.14552. arXiv:2305.14552 [cs].

Yusuf Mehdi. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web, 2023. URL https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. Investigating Saturation Effects in Integrated Gradients, October 2020. URL http://arxiv.org/abs/2010.12697. arXiv:2010.12697 [cs].

Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Towards Transparent and Explainable Attention Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4206–4216, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.387. URL https://aclanthology.org/2020.acl-main.387.

Hosein Mohebbi, Ali Modarressi, and Mohammad Taher Pilehvar. Exploring the Role of BERT Token Representations to Explain Sentence Probing Results, September 2021. URL http://arxiv.org/abs/2104.01477. arXiv:2104.01477 [cs].

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition, December 2015. URL http://arxiv.org/abs/1512.02479. arXiv:1512.02479 [cs, stat].

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. Measuring and improving faithfulness of attention in neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2791–2802, 2021.

Jesse Mu and Jacob Andreas. Compositional Explanations of Neurons, February 2021. URL http://arxiv.org/abs/2006.14032. arXiv:2006.14032 [cs, stat].

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

Michael Neely, Stefan F Schouten, Maurits JR Bleeker, and Ana Lucic. Order in the court: Explainable ai methods prone to disagreement. *arXiv preprint arXiv:2105.03287*, 2021.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. https://distill.pub/2020/circuits/zoom-in/, March 2020a. [Accessed 24-11-2023].

Chris Olah, Nick Cammarata, Chelsea Voss, Ludwig Schubert, and Gabriel Goh. Naturally Occurring Equivariance in Neural Networks — distill.pub. https://distill.pub/2020/circuits/equivariance/, December 2020b. [Accessed 27-11-2023].

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context

Learning and Induction Heads — transformer-circuits.pub. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html, March 2022. [Accessed 27-11-2023].

OpenAI. Language models can explain neurons in language models, 2023a. URL https://openai.com/research/language-models-can-explain-neurons-in-language-models?s=09.

OpenAI. Gpt-4 technical report, 2023b.

Cheonbok Park, Inyoup Na, Yongjang Jo, Sungbok Shin, Jaehyo Yoo, Bum Chul Kwon, Jian Zhao, Hyungjong Noh, Yeonsoo Lee, and Jaegul Choo. SANVis: Visual Analytics for Understanding Self-Attention Networks, September 2019. URL http://arxiv.org/abs/1909.09595. arXiv:1909.09595 [cs].

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1179. URL https://aclanthology.org/D18-1179.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language Models as Knowledge Bases?, September 2019. URL http://arxiv.org/abs/1909.01066. arXiv:1909.01066 [cs].

Michael Petrov, Chelsea Voss, Ludwig Schubert, Nick Cammarata, Gabriel Goh, and Chris Olah. Weight Banding — distill.pub. https://distill.pub/2020/circuits/weight-banding/, April 2021. [Accessed 27-11-2023].

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. Receval: Evaluating reasoning chains via correctness and informativeness. *arXiv preprint arXiv:2304.10703*, 2023.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

Luyu Qiu, Yi Yang, Caleb Chen Cao, Jing Liu, Yueyuan Zheng, Hilary Hei Ting Ngai, Janet Hsiao, and Lei Chen. Resisting Out-of-Distribution Data Problem in Perturbation of XAI, July 2021. URL http://arxiv.org/abs/2107.14000. arXiv:2107.14000 [cs].

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, et al. Question decomposition improves the faithfulness of model-generated reasoning. *arXiv preprint arXiv:2307.11768*, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4932–4942, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1487. URL https://aclanthology.org/P19-1487.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. On the Systematicity of Probing Contextualized Word Representations: The Case of Hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pp. 88–102, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.starsem-1.10.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation, July 2023. URL http://arxiv.org/abs/2307.11019. arXiv:2307.11019 [cs].

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, August 2016. URL http://arxiv.org/abs/1602.04938. arXiv:1602.04938 [cs, stat].

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

Alexis Ross, Ana Marasović, and Matthew E Peters. Explaining nlp models via minimal contrastive editing (mice). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3840–3852, 2021.

Soumya Sanyal and Xiang Ren. Discretized Integrated Gradients for Explaining Language Models, August 2021. URL http://arxiv.org/abs/2108.13654. arXiv:2108.13654 [cs].

Naomi Saphra. Interpretability Creationism. https://nsaphra.github.io/post/creationism/, 2022. [Accessed 22-10-2023].

Sofia Serrano and Noah A. Smith. Is Attention Interpretable?, June 2019. URL http://arxiv.org/abs/1906.03731. arXiv:1906.03731 [cs].

Lloyd S Shapley et al. A value for n-person games. 1953.

Yaozong Shen, Lijie Wang, Ying Chen, Xinyan Xiao, Jing Liu, and Hua Wu. An Interpretability Evaluation Benchmark for Pre-trained Language Models, July 2022. URL http://arxiv.org/abs/2207.13948. arXiv:2207.13948 [cs].

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. Integrated Directional Gradients: Feature Interaction Attribution for Neural NLP Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 865–878, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.71. URL https://aclanthology.org/2021.acl-long.71.

Chandan Singh, Aliyah R Hsu, Richard Antonello, Shailee Jain, Alexander G Huth, Bin Yu, and Jianfeng Gao. Explaining black box text modules in natural language with language models. *arXiv preprint arXiv:2305.09863*, 2023.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. Probing for Referential Information in Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4177–4189, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.384. URL https://aclanthology.org/2020.acl-main.384.

Joe Stacey, Yonatan Belinkov, and Marek Rei. Supervising model attention with human explanations for robust natural language inference. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 11349–11357, 2022.

Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. Seq2Seq-Vis: A Visual Debugging Tool for Sequence-to-Sequence Models, October 2018. URL http://arxiv.org/abs/1804.09299. arXiv:1804.09299 [cs].

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *International Conference on Machine Learning (ICML)*, 2017.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? Probing for sentence structure in contextualized word representations. 2019b.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Generating token-level explanations for natural language inference. *arXiv preprint arXiv:1904.10717*, 2019.

Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. Intrinsic Probing through Dimension Selection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 197–216, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.15. URL https://aclanthology.org/2020.emnlp-main.15.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama-2: Open foundation and fine-tuned chat models. 2023b. URL https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/.

Marcos Treviso, Alexis Ross, Nuno M Guerreiro, and André FT Martins. Crest: A joint framework for rationalization and counterfactual text generation. *The 61th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*, 2023.

Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. How Does BERT Answer Questions?: A Layer-Wise Analysis of Transformer Representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1823–1832, Beijing China, November 2019. ACM. ISBN 978-1-4503-6976-3. doi: 10.1145/3357384.3358028. URL https://dl.acm.org/doi/10.1145/3357384.3358028.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29, 2016.

Jesse Vig. BertViz: A Tool for Visualizing Multi-Head Self-Attention in the BERT Model. 2019.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580.

Elena Voita, Rico Sennrich, and Ivan Titov. Analyzing the Source and Target Contributions to Predictions in Neural Machine Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1126–1140, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.91. URL https://aclanthology.org/2021.acl-long.91.

Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch Specialization — distill.pub. https://distill.pub/2020/circuits/branch-specialization/, April 2021. [Accessed 26-11-2023].

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *International Conference on Learning Representations (ICLR)*, 2019.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*, 2022a.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. SemAttack: Natural Textual Attacks via Different Semantic Spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 176–205, Seattle, United States, 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.14. URL https://aclanthology.org/2022.findings-naacl.14.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. Simple synthetic data reduces sycophancy in large language models, August 2023a. URL http://arxiv.org/abs/2308.03958. arXiv:2308.03958 [cs].

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023b.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation, September 2019. URL http://arxiv.org/abs/1908.04626. arXiv:1908.04626 [cs].

Skyler Wu, Eric Meng Shen, Charumathi Badrinath, Jiaqi Ma, and Himabindu Lakkaraju. Analyzing chain-of-thought prompting in large language models via gradient-based feature attributions. 2023a.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6707–6723, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.523. URL https://aclanthology.org/2021.acl-long.523.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. Do PLMs Know and Understand Ontological Knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3080–3101, Toronto, Canada, 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.173. URL https://aclanthology.org/2023.acl-long.173.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. From language modeling to instruction following: Understanding the behavior shift in llms after instruction tuning. *arXiv preprint arXiv:2310.00492*, 2023c.

Zhengxuan Wu and Desmond C. Ong. On Explaining Your Explanations of BERT: An Empirical Study with Sequence Classification, January 2021. URL http://arxiv.org/abs/2101.00196. arXiv:2101.00196 [cs].

Zhengxuan Wu, Thanh-Son Nguyen, and Desmond Ong. Structured Self-Attention Weights Encode Semantics in Sentiment Analysis. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 255–264, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.24. URL https://aclanthology.org/2020.blackboxnlp-1.24.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4166–4176, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.383. URL https://aclanthology.org/2020.acl-main.383.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

Xi Ye and Greg Durrett. The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*, 35:30378–30392, 2022.

Xi Ye and Greg Durrett. Explanation selection using unlabeled data for chain-of-thought prompting, 2023.

Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. AttentionViz: A Global View of Transformer Attention, May 2023. URL http://arxiv.org/abs/2305.03210. arXiv:2305.03210 [cs].

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Jason Wu. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. *arXiv preprint arXiv:2306.01150*, 2023.

Yordan Yordanov, Vid Kocijan, Thomas Lukasiewicz, and Oana-Maria Camburu. Few-Shot Out-of-Domain Transfer Learning of Natural Language Explanations in a Label-Abundant Setup. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3486–3501, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.255.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc Concept Bottleneck Models, February 2023. URL http://arxiv.org/abs/2205.15480. arXiv:2205.15480 [cs, stat].

Lining Zhang, Mengchen Wang, Liben Chen, and Wenxin Zhang. Probing GPT-3's Linguistic Knowledge on Semantic Tasks. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 297–304, Abu Dhabi, United Arab Emirates (Hybrid), December 2022a. Association for Computational Linguistics. URL https://aclanthology.org/2022.blackboxnlp-1.24.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022b.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models, September 2023. URL http://arxiv.org/abs/2309.01219. arXiv:2309.01219 [cs].

Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual Probing Is [MASK]: Learning vs. Learning to Recall, December 2021. URL http://arxiv.org/abs/2104.05240. arXiv:2104.05240 [cs].

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.