Post-hoc Interpretability for Neural NLP: A Survey

ANDREAS MADSEN, Mila & Polytechnic Montreal
SIVA REDDY, Mila & McGill
SARATH CHANDAR, Mila & Polytechnique Montreal

Neural networks for NLP are becoming increasingly complex and widespread, and there is a growing concern if these models are responsible to use. Explaining models helps to address the safety and ethical concerns and is essential for accountability. Interpretability serves to provide these explanations in terms that are understandable to humans. Additionally, post-hoc methods provide explanations after a model is learned and are generally model-agnostic. This survey provides a categorization of how recent post-hoc interpretability methods communicate explanations to humans, it discusses each method in-depth, and how they are validated, as the latter is often a common concern.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Neural networks;**

Additional Key Words and Phrases: Interpretability, transparency, post-hoc explanations

ACM Reference format:
Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc Interpretability for Neural NLP: A Survey. *ACM Comput. Surv.* 55, 8, Article 155 (December 2022), 42 pages.
https://doi.org/10.1145/3546577

1 INTRODUCTION

Large neural NLP models, most notably BERT-like models [20, 36, 70], have become highly widespread, both in research and industry applications [133]. This increase of model complexity is motivated by a general correlation between model size and test performance [20, 56]. Due to their immense complexity, these models are generally considered black-box models. A growing concern is therefore if it is responsible to deploy these models.

Concerns such as safety, ethics, and accountability are particularly important when machine learning is used for high-stakes decisions, such as healthcare, criminal justice, finance, and so on. [100], including NLP-focused applications such as translation, dialog systems, resume screening, search, and so on. [38]. For many of these applications, neural models have been shown to exhibit unwanted biases and similar ethical issues [16, 20, 42, 75, 82, 100].

A. Madsen and S. Chandar also with École Polytechnique de Montréal.
S. Reddy also with McGill University.
S. Reddy also with Facebook CIFAR AI Chair.
S. Chandar also with Canada CIFAR AI Chair.
SC and SR are supported by the Canada CIFAR AI Chairs program and the NSERC Discovery Grant.
Authors' addresses: A. Madsen, Mila: 6666 Rue Saint-Urbain, Montréal, Quebec, Canada; email: andreas.madsen@mila.quebec; S. Reddy, Polytechnic Montreal: 2500 Chem. de Polytechnique, Montréal, QC H3T 1J4, Canada; email: siva.reddy@mila.quebec; S. Chandar, McGill: 845 Sherbrooke St W, Montreal, Quebec H3A 0G4, Canada; email: sarath.chandar@mila.quebec.

0360-0300/2022/12-ART155
https://doi.org/10.1145/3546577

Doshi-Velez and Kim [37] argue, among others [68], that these ethical and safety issues stem from an "incompleteness in the problem formalization". While these issues can be partially prevented with robustness and fairness metrics, it is often not possible to consider all failure modes. Therefore, quality assessment should also be done through model explanations. Furthermore, when models do fail in critical applications, explanations must be provided to facilitate the accountability process. Providing these explanations is often a core motivation for interpretability. In Section 2 we provide additional motivating factors.

Doshi-Velez and Kim [37] define *interpretability* as the "ability to explain or to present in understandable terms to a human". However, what constitutes as an "understandable" explanation is an interdisciplinary question. An important work from social science by Miller [78], argues that *effective explanations* must be selective in the sense one must select "one or two causes from a sometimes infinite number of causes". Such observation necessitates organizing interpretability methods by how and what they selectively communicate.

This survey presents such an organization in Table 1, where each row represents a communication approach. For example, the first row describes *input feature* explanations that communicate what tokens are most relevant for a prediction. In general, each row is ordered by how abstract the communication approach is, although this is an approximation. Organizing by the method of communication is discussed further in Section 1.1.

Each interpretability method uses different kinds of information to produce its explanation, in Table 1 this is indicated by the columns.[1] The columns are ordered by an increasing level of information. Again, this is an inexact ranking but serves as a useful tool to contrast the methods.

Table 1 frames the overall structure of this survey. Where each method section from 6 to 15 covers a row of Table 1. However, first we cover motivation (Section 2), how to validate interpretability (Section 4), and a motivating example (Section 3). The method sections can be read somewhat independently but will refer back to these general topics.

In contrast to other surveys and tutorials on interpretability methods [14, 15, 17, 25, 35, 79, 113, 119, 125] which only discusses the most popular approaches (usually 3 among *input features*, *adversarial examples*, *influential examples*, *projection*, and *linguistic information*), this survey offers a more diverse overview of communication approaches. We hope this leads to more questioning about how we communicate. Additionally, we consistently comment on how each method is validated (*groundedness*), an important discussion we find is often missing.

Finally, the survey limits itself to *post-hoc* interpretability methods. These are methods that provide their explanation after a model is trained and are often model-agnostic. This is in contrast to *intrinsic* methods, where the model architecture itself helps to provide the explanation. These terms are described further in Section 1.2.

1.1 Organizing by Method of Communication

As a categorization of communication strategies, it's standard in the interpretability literature to distinguish between methods that explain a single observation, called *local explanations*, and methods that explain the entire model called *global explanations* [2, 17, 24, 27, 37, 79]. In this survey, we also consider an additional category of methods that explains an entire output-class, which we call *class explanations*.

[1] *Black-box*: the method only evaluates the model. *Dataset*: the method has access to all training and validation observations. *Gradient*: the gradient of the model is computed. *Embeddings*: the method uses the word embedding matrix. *White-box*: the method knows everything about the model, such as all weights and all operations. However, the method is not specific to a particular architecture. *Model specific*: the method is specific to the architecture. Note, neural model in NLP are usually differentiability and have an embedding matrix. We, therefore, do not consider these properties constraints.

Table 1. Overview of *post-hoc* Interpretability Methods, where § Indicates the Section the Method is Discussed

		less information → more information					
		post-hoc					intrinsic
		black-box	dataset	gradient	embeddings	white-box	model specific
	local explanation						
lower abstraction	input features	SHAP § A.2	LIME § 6.2, Anchors § A.3	Gradient § 6.1, IG § A.1			Attention
	adversarial examples	SEA$^{\mathcal{M}}$ § B.1		HotFlip § 7.1			
	influential examples		Influence Functions$^{\mathcal{H}}$ § 8.1, TracIn^{C} § 8.3		Representer Pointers† § 8.2		Prototype Networks
	counter-factuals	Polyjuice$^{\mathcal{M},\mathcal{D}}$ § C.1	MiCE$^{\mathcal{M}}$ § 9.1				
	natural language	CAGE$^{\mathcal{M},\mathcal{D}}$ § 10.1					GEF$^{\mathcal{D}}$, NILE$^{\mathcal{D}}$
	class explanation						
	concepts					NIE$^{\mathcal{D}}$ § 11.1	
	global explanation						
	vocabulary				Project § 12.1, Rotate § 12.2		
higher abstraction	ensemble	SP-LIME § 13.1					
	linguistic information	Behavioral Probes$^{\mathcal{D}}$ § 14.1			Structural Probes$^{\mathcal{D}}$ § 14.2	Structural Probes$^{\mathcal{D}}$ § 14.2	Auxiliary Task$^{\mathcal{D}}$
	rules	SEAR$^{\mathcal{M}}$ § 15.1		Compositional Explanations of Neurons† § D.1			

Rows describe how the explanation is communicated, while columns describe what information is used to produce the explanation. The order of both rows and columns indicates a level of abstraction and amount of information, respectively. However, this order is only approximate.
Furthermore, because this survey focuses on *post-hoc* methods, the *intrinsic* section of this table is incomplete and merely meant to provide a few comparative examples. The specifc *intrinsic* methods shown are: *Attention* [9], *GEF* [69], *NILE* [63]. *Prototype Networks* and *Auxiliary Task* refer to types of models.
^{C}: Depends on checkpoints during training. $^{\mathcal{D}}$: Depends on supplementary dataset. $^{\mathcal{H}}$: Depends on second-order derivative. $^{\mathcal{M}}$: Depends on the supplementary model. †: Depends only on the dataset and white-box access.

To subdivide these categories further, Table 1 orders each communication strategy by their abstraction level. As an example, see Figure 1, where an *input features* explanation highlights the input tokens that are most responsible for a prediction; because this must refer to specific tokens, its ability to provide abstract explanations is limited. For a highly abstract explanation, consider the *natural language* category which explains a prediction using a sentence and can therefore use abstract concepts in its explanation.

Communication methods that have a higher abstraction level are typically easier to understand (more *human-grounded*), but the tradeoff is that they may reflect the model's behavior less (less *functionally-grounded*). Because the purpose of interpretability is to communicate the model to a human, this tradeoff is necessary [78, 100]. Which communication strategy should be used must be decided by considering the applications and to whom the explanation is communicated to. In Section 4 we discuss *human-groundedness* and *functionally-groundedness* in-depth and how to measure them, such that an informed decision can be made.

Table 1 does have some limitations. Firstly, ordering explanation methods by their abstraction level is an approximation, and while *global explanations* are generally more abstract than

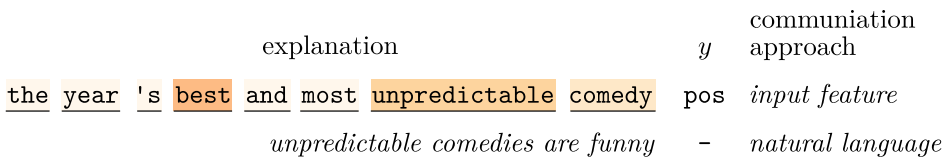

Fig. 1. Fictive visualization of an *input features* explanation, which highlights tokens and a *natural language* explanation, applied on a sentiment classification task [126]. y = pos means the gold label is *positive* sentiment.

local explanations this is not always true. For example, the explanation "simply print all weights" (not included in Table 1), is arguably the lowest possible abstraction level, however, it's also a *global explanation*. Secondly, there are explanation categories that are not included, such as *intermediate representations*. This category of explanation depends on models that are *intrinsically* interpretable, which are not the subject of this article. We elaborate on this in Section 1.2.

1.2 Intrinsic Versus Post-hoc Interpretability

A fundamental motivation for interpretability is accountability. For example, if a predictive mistake happens which caused harm, it is important to explain why this mistake happened [38]. Similarly, for high-stakes decisions, it is important to minimize the risk of model failure by explaining the model before deployment [100]. In other words, it is important to distinguish between when interpretability is applied proactively or retroactively to the model's deployment.

It is standard in the literature to categorize if an interpretability method can be applied retroactively or proactively. Unfortunately, the terminology for this taxonomy is not standardized [24]. This survey focuses on the methods that can be applied retroactively, for which the term *post-hoc* is used. Similarly, we use the term *intrinsic* to refer to models that are interpretable by design. These terms were chosen as the best compromise between established terminology [39, 53, 79] and correctness in terms of their dictionary definition.

Intrinsic *methods.* These inherently depend on models that by design are interpretable. Because of this relation, it is also often referred to as *white-box models* [27, 39, 100]. However, the term *white-box* is slightly misleading, as it is often only a part of the transparent model.

As an example, consider *intermediate representation* explanations, this category depends on a model that is constrained to produce a meaningful *intermediate representation*. In Neural Modular Networks [7, 46] this could be `find-max-num(filter(find()))`, which represents how to extract an answer from a question-paragraph-pair. However, how this representation is produced is not necessarily *intrinsically* interpretable.

Intrinsic methods are attractive because they may be more responsible to use in high-stakes decision processes. However, as Jacovi and Goldberg [53] argue, "a method being *inherently interpretable* is merely a claim that needs to be verified before it can be trusted". Verifying this is often non-trivial, as has repeatedly been shown with *Attention* [9], where multiple articles have found contradicting conclusions regarding interpretability [54, 104, 121, 129].

Post-hoc *methods.* These are the focus of this article. While many *post-hoc* methods are model-agnostic, this is not a necessary property, and in some cases does only apply to a category of models. Indeed, in this article, only methods that apply to neural networks are discussed.

Because of the inherent ability to explain the model after training, *post-hoc* methods are valuable in legal proceedings, where models may need to be explained retroactively [38]. Additionally, they fit into existing quality assessment structures, such as those used to regulate banking, where quality assessment is also done after a model has been built [17]. Finally, it is guaranteed that they will not affect model performance.

However, *post-hoc* methods are often criticized for providing false explanations, and it has been questioned if it is reasonable to expect models, that were not designed to be explained, to be explained anyway [100]. This is a valid concern, however producing *intrinsic* methods is often very task-dependent and therefore a difficult process that is rarely done in the industry [17]. *Post-hoc* methods are often much more adaptable and their impact can therefore be much greater if they can provide accurate explanations. The question of how to validate explanations is therefore very important and is covered in detail in Section 4. Furthermore, we pay special attention to how each method is validated in the literature throughout the survey.

Comparing. Both *Intrinsic* and *post-hoc* methods have their merits, but often provide different values in terms of *accountability*. Finally, *post-hoc* methods can often be applied also to *intrinsically* interpretable models. Observing a correlation between methods from these two categories can therefore provide validation of both methods [54].

2 MOTIVATIONS FOR INTERPRETABILITY

The need for interpretability comes primarily from an "incompleteness in the problem formalization" [37], meaning if the model was constrained and optimized to prevent all possible ethical issues, interpretability would be much less relevant. However, because perfect optimization is unlikely, hence *safety* and *ethics* are strong motivations for interpretability.

Additionally, when models misbehave there is a need for explanations, to hold people or companies accountable, hence *accountability* is often a core motivation for interpretability. Finally, explanations are often useful, or sometimes necessary, for gaining *scientific understanding* about models. This section aims at elaborating on what exactly is understood by these terms and how interpretability can address them.

Ethics, in the context of interpretability, is about ensuring that the model's behavior is aligned with common ethical and moral values. Because there does not exist an exact measure for this desideratum, this is ultimately something that should be judged qualitatively by humans, for example by an *ethics review committee*, who will inspect the model explanations.

For some ethical concerns, such as discrimination, it may be possible to measure and satisfy this ethical concern via fairness metrics and debiasing techniques [42]. However, this often requires a finite list of protected attributes [50], and such a list will likely be incomplete, hence the need for a qualitative assessment [37, 68].

Safety, is about ensuring the model performs within expectations in deployment. As it is nearly impossible to truly test the model, in the end-to-end context that it will be deployed, ensuring *safety* does to some extent involve qualitative assessment [37]. Lipton [68] frames this as *trust*, and suggests one interpretation of this is "that we feel comfortable relinquishing control to it".

While all types of interpretability can help with *safety*, in particular, *adversarial examples* and *counterfactuals* are useful, as they evaluate the model on data outside the test distribution. Lipton [68] frames this in the broader context of *transferability*, which is the model's robustness to adversarial attacks and distributional shifts.

Accountability, relates to explaining the model when it does fail in production. The "right to explanation", regarding the logic involved in the model's prediction, is increasingly being adopted, most notably in the EU via its GDPR legislation. However, also the US and UK have expressed support for such regulation [38]. Additionally, industries such as banking, are already required to audit their models [17].

		$p(y\|\mathbf{x};\theta)$	y
x	the year 's best and most unpredictable comedy	0.91	pos
x	we never feel anything for these characters	0.95	neg
x	handsome but unfulfilling suspense drama	0.18	neg

Fig. 2. Three examples from the SST dataset [109]. $\mathbf{x}$ is the input, with each token denoted by an underline. y is the gold target label, where pos is *positive* and neg is *negative* sentiment. Finally, $p(y|\mathbf{x})$ is the model's estimate of $\mathbf{x}$ belonging to category y. Note that the model predicts the 3rd (last) wrong, indicated with red.

Accountability is perhaps the core motivation of interpretability, as Miller [78] writes "Interpretability is the degree to which a human can understand the cause of a decision", and it is exactly the causal reasoning that is relevant in *accountability* [38].

Scientific Understanding, addresses a need from researchers and scientists, which is to generate hypotheses and knowledge. As Doshi-Velez and Kim [37] frames it, sometimes the best way to start such a process is to ask for explanations. In model development, explanations can also be useful for *model debugging* [17], which is often facilitated by the same kinds of explanations.

3 MOTIVATING EXAMPLE

Because *post-hoc* methods are often model-agnostic, explaining and discussing them can often become abstract. To make the method sections as concrete and comparable as possible this survey will show fictive examples often based on the "**Stanford Sentiment Treebank**" (**SST**) dataset [109]. The SST dataset has been modeled using LSTM [126], Self-Attention-based models [36], and so on, all of which are popular examples of neural networks.

We use a sequence-to-class problem because this is what most interpretability methods applies to. Although some are agnostic to the problem type and others are specific to sequence-to-sequence problems. Throughout this survey, we attempt to highlight what problems each method applies to.

The model responsible for the predictions in Figure 2 can be explained by asking different questions, each of which communicates a different aspect of the model that is covered in the sections of this survey. Sometimes these explanation relates to a single observation, other times the explanation relates to the whole model.

local explanations. explain a single observation:

Input Features	*Which tokens are most important for the prediction, Section 6.*
Adversarial Examples	*What would break the model's prediction, Section 7.*
Influential Examples	*What training examples influenced the prediction, Section 8.*
Counterfactuals	*What does the model consider a valid opposite example, Section 9.*
Natural Language	*What would a generated natural language explanation be, Section 10.*

Class explanations. summarize the model, but only with regard to one selected class:

Concepts	*What concepts (e.g., movie genre) can explain a class, Section 11.*

Global explanations. summarize the entire model with regards to a specific aspect:

Vocabulary	*How does the model relate words to each other, Section 12.*
Ensemble	*What examples are representative of the model, Section 13.*
Linguistic information	*What linguistic information does the model use, Section 14.*
Rules	*Which general rules can summarize an aspect of the model, Section 15.*

4 MEASURES OF INTERPRETABILITY

Because interpretability is by definition about explaining the model to humans [37, 78], and these explanations are often qualitative, it is not clear how to quantitatively evaluate and compare interpretability methods. This ambiguity has lead to much discussion. Most notable is the *intrinsically* interpretable method *Attention*, where different measures of interpretability have been published resulting in conflicting findings [54, 104, 129].

In general, there is no consensus on how to measure interpretability. However, validation is still paramount. As such, this section attempts to cover the general categories, themes, and methods that have been proposed. Additionally, each method section, starting from *input features*, in Section 6, will briefly cover how the authors choose to evaluate their method.

To describe the evaluation strategies, we use the terminology defined by Doshi-Velez and Kim [37], which separates the evaluation of interpretability into three categories, *functionally-grounded*, *human-grounded*, and *application-grounded*. This categorization reflects the need to have explanations that are useful to humans (*human-grounded*) and accurately reflect the model (*functionally-grounded*).

Application-grounded. evaluation is when the interpretability method is evaluated in the environment it will be deployed. For example, does the explanations result in higher survival-rates in a medical setting, a higher-grades in a homework-hint system, or a better model in a label-correction setting [37, 132]. Importantly, this evaluation should include the baseline where the explanations are provided by humans.

Due to the application-specific and time-consuming nature of this approach, *application-grounded* evaluation is rarely done in NLP interpretability research. Instead, more synthetic and general evaluation setups can be used, which is what *functionally-grounded* and *human-grounded* evaluation is about. These categories each provide an important but different aspect for validating interpretability and should therefore be used in combination.

Human-grounded. evaluation checks if the explanations are useful to humans. Unlike *application-grounded*, the task is often simpler and the task itself can be evaluated immediately. Additionally, expert humans are often not required [37]. In other literature, this is known as *simulatability* [68] and *comprehensibility* [97].

Although, *human-grounded* evaluation is much more efficient than *application-grounded* evaluation, the human aspect still takes time. An unfortunate but common approach is therefore to replace the human with a simulated user. This is unfortunate as providing explanations that are informative to humans is a non-trivial task, and often involves interdisciplinary knowledge from the **human-computer interaction** (**HCI**) and social science fields. Replacing a human with a simulated user, therefore leads to over-optimistic results.

Miller [78] provides an excellent overview on what effective explanation is from the social science perspective, and criticizes current works by saying "most work in explainable artificial intelligence uses only the researchers' intuition of what constitutes a "good" explanation".

It is therefore critical that interpretability methods are *human-grounded*. These are common strategies to measure *human-grounded*, used both in NLP and other fields:

- Humans have to choose the best model based on an explanation [94].
- Humans have to predict the model's behavior on new data [92].
- Humans have to identify an outlier example called an intruder [26]. While it can be used on other fields, it is most common in NLP where it is used with *vocabulary* explanations [84].

Functionally-grounded. evaluation checks how well the explanation reflects the model. This is more commonly known as *faithfulness* [39, 53, 94, 129] or sometimes *fidelity* [97].

It might seem surprising that an explanation, which is directly produced from the model, would not reflect the model. However, even intrinsically interpretable methods such as *Attention* and *Neural Modular Networks* have been shown to not reflect the model [54, 112].

Interestingly, *human-grounded* interpretability methods can not reflect the model perfectly, because humans require explanations to be selective, meaning the explanation should select "one or two causes from a sometimes infinite number of causes" [78]. Regardless, the explanations must still reflect the model to some extent, which surprisingly is not always the case [53, 100]. Additionally, explanations that provide a similar type of explanation, with similar selectiveness, should compete on proving the explanation that best reflects the model.

For some tasks, measuring if an interpretability method is *functionally-grounded* is trivial. In the case of *adversarial examples*, it is enough to show that the prediction changed and the adversarial example is a paraphrase. In other cases, most notably *input features*, providing a *functionally-grounded* metric can be very challenging [3, 51, 53, 60, 135].

In general, common evaluation strategies, both in NLP and other fields, are

- Comparing with an intrinsically interpretable model, such as logistic regression [94].
- Comparing with other post-hoc methods [54].
- Proposing axiomatic desirables [114].
- Benchmarking against random explanations [51, 73].

5 METHODS OF INTERPRETABILITY

The main objective of this survey is to give an overview of *post-hoc* interpretability methods and categorize them by how they communicate. Section 6 to Section 15 will be dedicated toward this goal.

Table 1 represents a table-of-content, relating each section to a communication approach, but also contrasts the different methods by what information they use. In addition, the motivating example in Section 3 gives a brief idea of the different communication approaches.

Each method section from *input features* (Section 6) to *rules* (Section 15) covers one communication approach, corresponding to one row in Table 1, and can be read somewhat independently. Each section discusses the purpose of the communication approach and covers the most relevant methods and how they are evaluated. Because interpretability is a large field, this survey chooses methods based on historical progression and diversity regarding what information they use, this is discussed more in *limitations* (Section 16). Finally, at the end of each method section, we discuss the general trends and issues related to that communication approach.

Each method section will use the terminology[2] covered in *motivation for interpretability* (Section 2) and *measures of interpretability* (Section 4).

6 INPUT FEATURES

An *Input feature* explanation is a *local explanation*, where the goal is to determine how important an *input feature*, e.g., a token, is for a given prediction. This approach is highly adaptable to different problems, as the input features are always known and are often meaningful to humans. Especially in NLP, the input features will often represent words, sub-words, or characters. Knowing which words are the most important, can be a powerful explanation method. An *input feature* explanation of the input $\mathbf{x}$, is represented as

$$\mathbf{E}(\mathbf{x}, c) : \mathrm{I}^{\mathbf{d}} \rightarrow \mathbb{R}^{\mathbf{d}}, \text{ where I is the input domain and } \mathbf{d} \text{ is the input dimensionality.} \quad (1)$$

[2]If you are viewing this article in a PDF reader, each term will *link* to the section where it's defined.

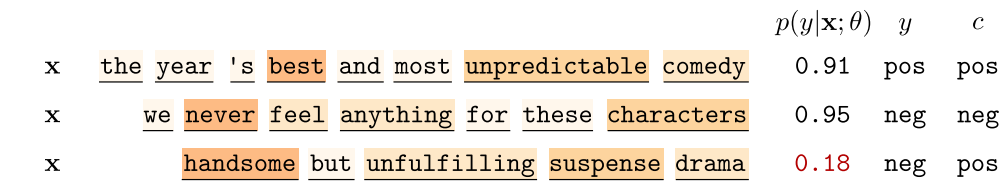

Fig. 3. Hypothetical visualization of applying $\mathbf{E}_{\text{gradient}}(\mathbf{x})$, where c is the explained class. Note that because the vocabulary dimension is reduced away, typically using the L^2-norm, it is not possible to separate positive influence (red) from negative influence (blue).

Note that, when the output is a score of importance the explanation is called an *importance measure*.

Importantly, *input feature* explanations can only explain one scalar, meaning one class at one timestep. In a sequence-to-sequence application, the explanation is therefore repeated for each time step [66, 72] although this may not respect the combinatorial complexities [5]. Additionally, the selected class is either the most likely class or the true-label class, in this section the explained class is denoted with c. For all methods in this section, except *Anchors*, c can be set as desired.

6.1 Gradient

One simple *importance measure*, is taking the gradient w.r.t. the input [8, 66].

$$\mathbf{E}_{\text{gradient}}(\mathbf{x}, c) = L_p(\nabla_{\mathbf{x}} p(c|\mathbf{x};\theta)), \text{where } L_p \in \{L_1, L_2, L_\infty\} \text{ and } p(c|\mathbf{x};\theta) \text{ is the model's probability output.} \tag{2}$$

This essentially measures the change of the output, given an ϵ-change to each input feature. Note, while NLP features are often discrete, it is still possible to take the gradient w.r.t. the one-hot-encoding by treating it as continuous. Although, because the one-hot-encoding has shape $\mathbf{x} \in \mathrm{I}^{V \times T}$, where V is the vocabulary size and T is the input length, it is necessary to reduce away the vocabulary dimension (often using an L_p-norm) such $\mathbf{E}(\mathbf{x}) \in \mathbb{R}^T$, when visualizing the importance per word as seen in Figure 3.

The primary argument for the *gradient* method being *functionally-grounded*, is that for a linear model $\mathbf{xW}$, the explanation would be $\mathbf{W}^\top_{c,:}$ which is clearly a valid explanation [3]. However, this does not guarantee *functionally-groundedness* for non-linear models, as the explanation mearly relates to a first-order Taylor approximation [66]. Additionally, areas of the input may be important but have zero gradients, this issue is discussed Section A.1.

Finally, it can be sensible to consider the scale of $\mathbf{x}$ too, hence the extension $\mathbf{x} \odot \nabla_{\mathbf{x}} p(c|\mathbf{x};\theta)$ is sometimes preferred. Although, a counter-argument is that $\mathbf{x}$ does not directly relate to the model, and this can therefore result in a less faithful explanation [3].

6.2 LIME

Another popular approach is *LIME* [94]. This distinguishes itself from the gradient-based methods by not relying on gradients. Instead, it samples nearby observations $\tilde{\mathbf{x}}$ and uses the model estimate $p(c|\tilde{\mathbf{x}})$ to fit a logistic regression. The parameters $\mathbf{w}$ of the logistic regression then represents the *importance measure*, since larger parameters would mean a greater effect on the output. An example of this can be seen in Figure 4.

$$\mathbf{E}_{\text{LIME}}(\mathbf{x}, c) = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^{k} (p(c|\tilde{\mathbf{x}}_i;\theta) \log(q(\tilde{\mathbf{x}}_i)) + (1 - p(c|\tilde{\mathbf{x}}_i;\theta)) \log(1 - q(\tilde{\mathbf{x}}_i)) + \lambda \|\mathbf{w}\|_1 \quad \text{where } q(\tilde{\mathbf{x}}) = \sigma(\mathbf{w}\tilde{\mathbf{x}}). \tag{3}$$

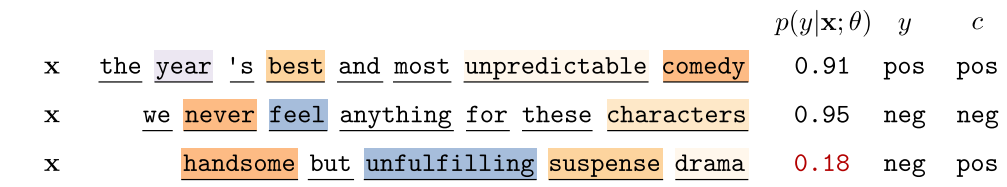

Fig. 4. A fictive visualization of LIME, where the weights of the logistic regression determine the *importance measure*. Note that for LIME, it is possible to have negative importance (indicated by blue). Furthermore, some tokens have no importance score, due to the L^1-regularizer.

One major complication of *LIME* is how to sample $\tilde{\mathbf{x}}$, representing the nearby observations. In the original article [94], they use a **Bag-Of-Words** (**BoWs**) representation with a cosine distance. While this approach remains possible with a model that works on sequential data, such distance metrics may not effectively match the model's internal space. In more recent work [134], they sample $\tilde{\mathbf{x}}$ by masking words of $\mathbf{x}$. However, this requires a model that supports such masking.

The advantages of *LIME* are that it only depends on black-box information and the dataset, therefore no gradient calculations are required. Secondly, it uses a LASSO logistic regression, which is a normal logistic regression with an L_1-regularizer. This means that its explanation is selective, as in sparse, which may be essential for providing a human-friendly explanation [78].

Ribeiro et al. [94] show that LIME is *functionally-grounded* by applying LIME on *intrinsically* interpretable models, such as a logistic regression model, and then compare the LIME explanation with the *intrinsic* explanation from the logistic regression. They also show *human-groundedness* by conducting a human trial experiment, where non-experts have to choose the best model, based on the provided explanation, given a "wrong classifier" tranined on a bias dataset and a "correct classifier" trained on a curated dataset.

6.3 Discussion

Groundedness. The *functionally-groundedness* of *input feature* explanations has received a lot of attention and discussion, however, there is still little consensus on what is *functionally-grounded* or how to even measure it [3, 11, 54, 60, 73, 104, 121, 129].

Future work. It has been suggested, that a general *functionally-grounded* post-hoc *input feature* explanation method just does not exists [100], an analogue to the no-free-lunch theorem. For this reason, a new trend in NLP is to develop architecture specific *input feature* explanations [1, 21], for example using attention. Although others are against this direction and do not think that attention can provide more *functionally-grounded* explanations than general alternatives [12].

Such high-level questions are likely difficult to answer without a more fundamental understanding of what the *functionally-groundedness* desirables are. We, therefore, advocate for continuing the effort in measuring *functionally-groundedness* but to focus more on establishing the fundamental desirables.

7 ADVERSARIAL EXAMPLES

An *adversarial example*, is an input that causes a model to produce a wrong prediction, due to limitations of the model. The adversarial example is often produced from an existing example, for which the model produces a correct prediction. Because the *adversarial example* serves as an explanation, in the context of an existing example it is a *local explanation*.

Wang et al. [127] provide a thorough survey on *adversarial example* explanations, and also goes in-depth regarding taxonomy, using *adversarial examples* for robustness, and similarity scores

between the existing example and the *adversarial example*. Additionally, the survey by Belinkov and Glass [15] also has a section on adversarial examples.

In this survey, we therefore focus on just two explanation methods. These *adversarial example* methods informs us about the support boundaries of a given example, which then informs us about the logic involved and therefore provides interpretability. In fact, this explanation can be similar to the *input feature* methods, discussed in Section 6. Many of those methods also indicate what words should be changed to alter the prediction. An important difference is that *adversarial* explanations are contrastive, meaning they explain by comparing with another example, while *input features* explain only concerning the original example. Contrastive explanations are, from a social science perspective, generally considered more *human-grounded* [78].

In the following discussions, we refer to the original example as $\mathbf{x}$ and the adversarial example as $\tilde{\mathbf{x}}$. The goal is to develop an adversarial method A, that maps from $\mathbf{x}$ to $\tilde{\mathbf{x}}$:

$$A(\mathbf{x}) \rightarrow \tilde{\mathbf{x}}. \tag{4}$$

Importantly, to ensure that an *adversarial example* method is *functionally-grounded*, one only needs to assert that the predicted label changes while the gold label remains the same. Additionally, it's a desirable to have the original and adversarial example to be similar, in many applications this can be framed as paraphrasing. Compared to other explanation types, these properties are reasonably trivial to measure. See Section 4 for a general discussion on measures of interpretability.

Finally, because *adversarial example* explanations are framed by the output class, these explanations do not generalize easily to sequence-to-sequence problems. Although one could imagine for example an offensive-text classifier, which reduces the sequence-to-sequence model back to a sequence-to-class model.

7.1 HotFlip

A great example of the relation between *input feature* explanations and *adversarial examples* is *HotFlip* [40]. Here the effect of changing token v to another token $\tilde{v}$ at position t, on the model loss $\mathcal{L}$, is estimated via using gradients

$$\mathcal{L}(y, \tilde{\mathbf{x}}_{t:v\rightarrow\tilde{v}}) - \mathcal{L}(y, \mathbf{x}; \theta) \approx \frac{\partial \mathcal{L}(y, \mathbf{x}; \theta)}{\partial x_{t,\tilde{v}}} - \frac{\partial \mathcal{L}(y, \mathbf{x}; \theta)}{\partial x_{t,v}}, \tag{5}$$

where $\tilde{\mathbf{x}}_{t:v\rightarrow\tilde{v}}$ is the one-hot-encoded input $\mathbf{x}$, with the token v at position t changed to $\tilde{v}$. Additionally, $x_{t,\tilde{v}}$ and $x_{t,v}$ are the scalar components of the one-hot-encoded input $\mathbf{x}$.

Had a gradient approximation not been used, the alternative would be to exactly compute a forward pass for every possible token swap. Instead, this approximation only requires one backward pass. To produce an adversarial sentence with multiple tokens changed, the authors use a beam-search approach. A visualization of *HotFlip* can be seen in Figure 5.

$$A_{\text{HotFlip}}(\mathbf{x}) = \underset{\tilde{\mathbf{x}}_{t:v\rightarrow\tilde{v}}}{\text{argmax}} \frac{\partial \mathcal{L}(y, \mathbf{x}; \theta)}{\partial x_{t,\tilde{v}}} - \frac{\partial \mathcal{L}(y, \mathbf{x}; \theta)}{\partial x_{t,v}}. \tag{6}$$

The *HotFlip* article [40] primarily investigates character-level models, for which the desire is to build a model that is robust against typos. However, in terms of word-level models, it is necessary to constrain the possible changes, such that the adversarial sentence is a paraphrase. They do this via the word-embeddings, such that the adversarial word and the original word are constrained to have a cosine similarity of at least 0.8.

The *HotFlip* approach has proven effective for other adversarial explanation methods, such as the aforementioned Universal Adversarial Triggers [124].

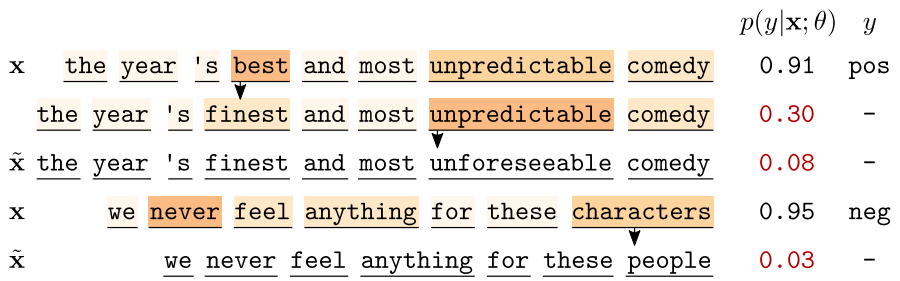

Fig. 5. Hypothetical visualization of *HotFlip*. The highlight indicates the gradient w.r.t. the input, which HotFlip uses to select which token to change. $\mathbf{x}$ indicates the original sentence, and $\tilde{\mathbf{x}}$ indicates the adversarial sentence.

7.2 Discussion

Groundedness. Adversarial example are as mentioned, easy to measure *functionally-groundedness* on and should be *human-grounded* due to their contrastive nature [78]. However, we are not aware of any work which explicitly tests for *human-groundedness*. This is likely because it is considered to be a given, but we advocate for testing such a hypothesis anyway.

Future work. The difficulty with *adversarial example* explanations lies in the search procedure. For example, *HotFlip* [40] uses a greedy sequential search algorithm and would therefore not be able to identify combinatorial effects like a double-negative. While *SEA* Ribeiro et al. [96] depends on an expensive paraphrase generation model.

One typical limitation of *adversarial example* methods is that they provide no control of the search direction. Hypothetically, while changing "unpredictable" to "unforeseeable" could provide the largest source of error due to a robustness issue, it might be more interesting to discover that changing "womans' chess club" to "mens' chess club" also flips the label. Unfortunately, this aspect is usually not considered because the motivation for *adversarial example* generation is often robustness and debiasing.

8 SIMILAR EXAMPLES

For a given *input example*, an *influential examples* explanation finds examples from the training dataset, which in terms of the model's understanding, looks like the *input example*. Because of this explanation method centers around a specific *input example* it is a *local explanation*. Note that is different from just an distance metric on the inputs, such as BLEU [83], as this does not depend on the model.

Influential examples explanations can be quite useful. For example, for discovering dataset artifacts as some of the *influential examples* may have nothing to do with the *input example*, except for the artifacts. Additionally, they are commonly used to discover mislabeled observations.

The *influential examples* can always be presented as just the examples and a similarity score, see Figure 6. Because the only presentation difference is the similarity score, this chapter does not include example figures for each method.

8.1 Influence Functions

Influence functions is a classical technique from robust statistics [32]. However, in robust statistics, there are strong assumptions regarding convexity, low-dimensionality, and differentiability. Recent efforts in deep learning remove the low-dimensionality constraint and to some extent the convexity constraint [61].

	x	$p(y\|\mathbf{x};\theta)$	y	Δ
x	the year 's best and most unpredictable comedy	0.91	pos	0.21
$\tilde{\mathbf{x}}$	a delightfully unpredictable , hilarious comedy	0.95	pos	3.82
$\tilde{\mathbf{x}}$	loud and thoroughly obnoxious comedy	0.98	neg	-1.51

Fig. 6. Fictive result showing the *influential examples* $\tilde{\mathbf{x}}$, in relation to the *input example* $\mathbf{x}$, showing both examples with positive and negative influence. Δ is the similarity score, the scale and range may depend on the specific method. Note, it is possible to measure the influence of an example on itself. This can be useful to identify mislabled observations, as such observations will be important for their own prediction.

The central idea in *influence functions*, is to estimate the effect on the loss $\mathcal{L}$, of removing the observation $\tilde{\mathbf{x}}$ from the dataset. The most influential examples are those where the loss changes the most. Let $\tilde{\theta}$ be the model parameters if $\tilde{\mathbf{x}}$ had not been included in the training dataset, then the loss difference can be estimated using

$$\mathcal{L}(y, \mathbf{x}; \tilde{\theta}) - \mathcal{L}(y, \mathbf{x}; \theta) \approx \frac{1}{n} \nabla_\theta \mathcal{L}(y, \mathbf{x}; \theta)^\top H_\theta^{-1} \nabla_\theta \mathcal{L}(\tilde{y}, \tilde{\mathbf{x}}; \theta). \tag{7}$$

Importantly, the Hessian H_θ needs to be positive-definite, which can only be guaranteed for convex models. The authors Koh and Liang [61] avoid this issue, by adding a diagonal to the Hessian, until it is positive-definite. Additionally, they solve the computational issue of computing an inverse Hessian, by formulating Equation (7) as an inverse Hessian-vector product. Such formulation can be approximated in $O(np)$ time, where n is the number of observations and p is the number of parameters, hence a computational complexity identical to one training epoch. Note however, that the inverse Hessian-vector product needs to be computed for every explained test observation $\mathbf{x}$.

One limitation of *influence functions* is that computing the *influence functions* is not always numerically stable [136], because Equation (7) uses the gradient $\nabla_\theta \mathcal{L}(\tilde{y}, \tilde{\mathbf{x}}; \theta)$ which is optimized to be close to zero.

Koh and Liang [61] looked at support-vector-machines, which are known to be convex, and convolutional neural networks which are generally non-convex. Han et al. [47] then extended the analysis of *influence functions* to BERT [36]. This is a crucial step, as BERT may be much further from convexity than CNNs, thus cause the *influence functions* to be less *functionally-grounded*.

Han et al. [47] validates for *functionally-groundedness* by removing the 10% most influential training examples from the dataset and then retrain the model. The results show a significant decrease in the model's performance on the test split, compared to removing the 10% least influential examples and 10% random examples, validating that the influential examples are important.

Additionally, Koh and Liang [61] measures *functionally-groundedness* by setting 10% of training observations to a wrong label. *Influence functions* is then used to select a fraction of the dataset, for which labels are corrected. The metric is then how many mislabeled observations were identified and the performance difference. The idea being, wrongly labeled observations should affect the loss more than correctly labeled observations, hence *influence functions* will tend to find wrongly labeled observations. Han et al. [47] perform a similar experiment, but instead removes observations based on importance and then measures the performance difference. Both experiments validates that *influence functions* are *functionally-grounded*.

Performance considerations. A criticism of influence functions has been that it is computationally expensive. Although $\nabla_\theta \mathcal{L}(y, \mathbf{x}; \theta)^\top H_\theta^{-1}$ can be cached for each test example, it is still too computationally intensive for real-time inspection of the model. Additionally, having to compute the

weight-gradient $\nabla_\theta \mathcal{L}(\tilde{y}, \tilde{\mathbf{x}}; \theta)$ and inner-product for every training observation, does not scale sufficiently. To this end, Guo et al. [45] propose to only use a subset of training data, using a KNN clustering. Additionally, they show that the hyperparameters when computing $\nabla_\theta \mathcal{L}(y, \mathbf{x}; \theta)^\top H_\theta^{-1}$ can be tuned to reduce the computation to less than half.

8.2 Representer Point Selection

An alternative to *influence functions*, is the Representer theorem [103]. The central idea is that the logits of a test example $\mathbf{x}$, can be expressed as a decomposition of all training samples $f(\mathbf{x}) = \sum_{i=1}^n \boldsymbol{\alpha}_i \kappa(\mathbf{x}, \tilde{\mathbf{x}}_i)$. The original Representer theorem [103] works on *reproducing kernel Hilbert spaces*, which is not applicable for deep learning. However, recent work has applied the idea to neural networks [136].

Let $\boldsymbol{\theta}_L$ be the weight matrix of the final layer, such that the logits $f(\mathbf{x}; \theta) = \boldsymbol{\theta}_L \mathbf{z}_{L-1}(\mathbf{x}; \theta)$, then if the regularized loss $\frac{1}{n} \sum_{i=1}^n \mathcal{L}(\tilde{y}_i, \tilde{\mathbf{x}}_i; \theta) + \lambda \|\boldsymbol{\theta}_L\|^2$, is a stationary point and $\lambda > 0$, then

$$f(\mathbf{x}) = \sum_{i=1}^n \boldsymbol{\alpha}_i \mathbf{z}_{L-1}(\tilde{\mathbf{x}}_i; \theta)^\top \mathbf{z}_{L-1}(\mathbf{x}; \theta), \text{where } \boldsymbol{\alpha}_i = \frac{1}{2\lambda \cdot n} \frac{\partial \mathcal{L}(\tilde{y}_i, \tilde{\mathbf{x}}_i; \theta)}{\partial \mathbf{z}_{L-1}(\mathbf{x}_i; \theta)} . \tag{8}$$

To understand the importance of each training observation $\tilde{\mathbf{x}}_i$, regarding the prediction of class c for the test example $\mathbf{x}$, one just looks at the c'th element of each term $\boldsymbol{\alpha}_i \mathbf{z}_{L-1}(\tilde{\mathbf{x}}_i; \theta)^\top \mathbf{z}_{L-1}(\mathbf{x}; \theta)$. This approach is more numerically stable than *influence functions* [136], but has the downside of only depending on the intermediate representation of the final layer, while *influence functions* employs the entire model.

Because *Representer Point Selection* does depend on a specific model setup, where the last layer is regularized, this could be considered an *intrinsic* method. However, Yeh et al. [136] show that the stationary solution can be achieved *post-hoc*, meaning after learning, with minimal impact on the model predictions. They do this via the optimization problem

$$\boldsymbol{\theta}_L = \underset{\boldsymbol{W}}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{i=1}^n \mathcal{L}(p(\cdot|\tilde{\mathbf{x}}_i; \theta), \boldsymbol{W} \mathbf{z}_{L-1}(\tilde{\mathbf{x}}_i; \theta)) + \lambda \|\boldsymbol{W}\|^2 \right), \tag{9}$$

where θ is the original model parameters, $\boldsymbol{\theta}_L$ are the new parameters for the last layer, and $\mathcal{L}$ is the full cross-entropy loss. Because this is a fairly low-dimensional problem, fine-tuning this can be done with an L-BFGS optimizer or similar [136].

Yeh et al. [136] show this method is *functionally-grounded* on a computer vision task, using a label-correction experiment similar to that in *influence functions*. In this case, $|\boldsymbol{\alpha}_{i,c}|$ is used to select the observations to perform label correction on. Their results show that *Representer Point Selection* and *influence functions* can identify wrong labels equally well, but that the observations which *Representer Point Selection* selects affects the models performance more. Unfortunately, Yeh et al. [136] do only show anecdotal results on an NLP task.

8.3 TracIn

The idea behind *TracIn* by Pruthi et al. [88] is to accumulate loss changes during training. Specifically, the loss change on the test observation $\mathbf{x}$ when optimizing $\tilde{\mathbf{x}}$. Pruthi et al. [88] first introduce an idealized version of this, which assumes optimization is done on one observation at a time (for example, SGD):

$$\text{TracInIdeal}(\tilde{\mathbf{x}}, \mathbf{x}) = \sum_{t \in \mathcal{T}_{\tilde{\mathbf{x}}}} \mathcal{L}(y, \mathbf{x}, \theta_t) - \mathcal{L}(y, \mathbf{x}, \theta_{t+1}), \text{where } \mathcal{T}_{\tilde{\mathbf{x}}} \text{ is timestep which optimized } \tilde{\mathbf{x}}. \tag{10}$$

TracIn TracIn is then a relaxation of this idealized version. Rather than using a direct loss difference, gradients are used. Rather than assuming stochastic gradient descent (or similar), mini-batches can be used. Rather than checking every time step, checkpoints collected during training can be used.

$$\text{TracIn}(\tilde{\mathbf{x}}, \mathbf{x}) = \frac{1}{b} \sum_{t \in C} \eta_t \nabla_{\theta_t} \mathcal{L}(y, \mathbf{x}, \theta_t) \cdot \nabla_{\theta_t} \mathcal{L}(\tilde{y}, \tilde{\mathbf{x}}, \theta_t), \tag{11}$$

where C is checkpoints, b is batch-size, and η_t is learning-rate.

Note, that the Equation (11) formulation is still based on plain gradient descent. However, Pruthi et al. [88] instruct how to adapt this to most learning algorithms (AdaGrad, Adam, Newton, and so on).

As a *functionally-grounded* evaluation, Pruthi et al. [88] repeat the label-correction experiment of *influence functions* and *Representer Point Selection*, and find that their method can better select mislabeled observations. Note that this was evaluated on CIFAR-10 and MNIST. Unfortunately, Pruthi et al. [88] does not do any evaluation on NLP tasks, but they do anecdotally show it works on an NLP application.

8.4 Discussion

Groundedness. Influential example explanations, is one of the few categories with a non-trival but appropriate *functionally-grounded* metric, namely the label-correction experiment, which is used somewhat consistently across articles. Unfortunately, this experiment has not been used on NLP tasks, and in general very little *functionally-grounded* validation has been done in NLP.

Additionally, the label-correction experiment is somewhat limited, as it evaluates the influence of a training observation on itself. This is not how a *Influential examples* explanation would be used in most applications, for example, dataset artifact discovery. We, therefore, suggest future work also include the experiment from Guo et al. [45] which uses information removal.

Future work. A natural question, when asking what training observations are influential is to also what part of them are important. *Influence functions* can answer this, although at an increased computational cost. However, *TracIn* can not. For sequential outputs, it is interesting to also be able to select parts of the output and ask what influenced this. Both of these questions, are becoming increasingly relevant with large-scale language models, where there is a large interest in understanding what caused a particular generation.

9 COUNTERFACTUALS

Counterfactual explanations are essentially answering the question "how would the input need to change for the prediction to be different?". Furthermore, these *counterfactual examples* should be a minimal-edit from the original example and fluent. However, all of these properties can also be said of *adversarial explanations*, and indeed some works confuse these terms. The critical difference is that *adversarial examples* should have the same gold label as the original example, while *counterfactual examples* should have a different gold label (often opposite) as the original example [99]. Because *Counterfactual explanations* are defined by the output class they are limited to sequence-to-class models.

Another common confusion is with *counterfactual datasets*, also known as *Contrast Sets*. These datasets are used in robustness research and could consist of *counterfactual examples*. However, these datasets are generated without using a model [41, 58], and can therefore not be used to explain the model. *Contrast Sets* are however important for ensuring a robust model.

In social sciences, *counterfactual explanations* are considered highly useful for a person's ability to understand causal connections. Miller [78] explains that "why" questions are often answered

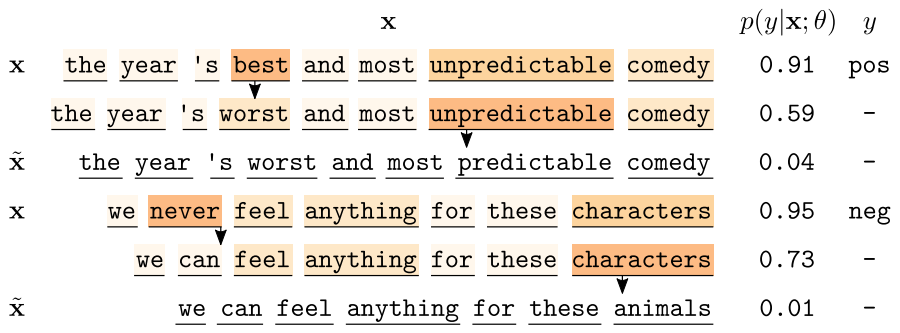

Fig. 7. Hypothetical visualization of how *MiCE* progressively creates a counterfactual $\tilde{\mathbf{x}}$ from an original sentence $\mathbf{x}$. The highlight shows the *gradient* $\nabla_{\mathbf{x}} f(\mathbf{x};\theta)_y$, which *MiCE* uses to know what tokens to replace.

by comparing *facts* with *foils*, where the term *foils* is the social sciences term for *counterfactual examples*.

9.1 MiCE

Like *Polyjuice* [134], *MiCE* [99] also uses an auxiliary model to generate *counterfactuals*. However, unlike *Polyjuice*, *MiCE* does not depend on auxiliary datasets and the counterfactual generation is more tied to the model being explained, rather than just using the model's predictions to filter the *counterfactual examples*.

The counterfactual generator is a T5 model [91], a sequence-to-sequence model, which is fine-tuned by input-output-pairs, where the input consists of the gold label and the masked sentence, while the output is the masking answer, see Equation (12) for an example.

$$\begin{aligned} \textit{input} &= \text{“}\texttt{label:}\ \underbrace{\text{positive}}_{\text{gold label}}\ \texttt{,}\ \texttt{input:}\ \underbrace{\text{This movie is [BLANK]!}}_{\text{masked sentence}}\text{”} \\ \textit{target} &= \text{“}\texttt{[CLR]}\ \underbrace{\text{really great}}_{\text{masking answer}}\ \texttt{[EOS]}\text{”} \end{aligned} \tag{12}$$

The *MiCE* approach to selecting which tokens to mask is to use an *importance measure*, specifically *the gradient w.r.t. the input*, and then mask the top x% most important consecutive tokens.

For generating counterfactuals, *MiCE* again masks tokens based on the *importance measure*, but then also inverts the gold label used for the T5-input Equation (12). This way the model will attempt to infill the mask, such that the sentence will have an opposite semantic meaning. This process is then repeated via a beam-search algorithm which stops when the model prediction changes, an example of this can be seen in Figure 7.

Because *MiCE* uses the model prediction to stop the beam-search, it will inherently be somewhat *functionally-grounded*. However, it may be that using the *gradient* as the *importance measure*, is not *functionally-grounded*. Ross et al. [99] validate that using the *gradient* is *functionally-grounded*, by looking at the number of edits and fluency of *MiCE* and compare it to a version of *MiCE* where random tokens are masked. They find that using the *gradient* significantly improves both fluency and reduces the number of edits it takes to change a prediction.

9.2 Discussion

Groundedness. While *counterfactual examples* are great for *human-grounded* explanation, they struggle with *functionally-groundedness*. The challenge comes from the desirables. On one side, a desirable is to provide a counterfactual example with the opposite gold label, an objective that is

independent of the model. Simultaneously the search procedure should be directed by the model behavior. These objectives can at times appear opposite, although *MiCE* provide a great example of how it can be done.

Future work. Because the motivation for *counterfactual examples* is often robustness, the search procedure often becomes only weakly dependent on the model such as *Polyjuice* or sometimes completly independent such as *Contrast Sets*.

While robustness is a perfectly valid research objective, we recommend being careful when using both robustness with interpretability to motivate the same method, as this often leads to *functionally-groundedness* issues. We would therefore advocate for more counterfactual research which focuses only on interpretability and *functionally-groundedness*.

10 NATURAL LANGUAGE

A common concern for many of the explanation methods presented in this survey is that they are difficult to understand for people without specialized knowledge. It is therefore attractive to directly generate an explanation in the form of *natural language*, which can be understood by simply reading the explanation for a given example. Because these utterances explain just a single example, they are a *local explanation*.

Most research in the area of *natural language* explanation uses the explanations to improve the predictive performance of the model itself. The idea is that by enforcing the model to reason about its behavior, the model can generalize better [23, 63–65, 69, 92]. These approaches are however in the category of *intrinsic* methods. While those methods are often quite general, they are not discussed in this survey which focuses on *post-hoc* methods.

These *post-hoc* methods are referred to as *rationalization* methods, in the sense that they attempt to explain after a prediction has been made [92]. Note that the term is a misnomer, as rationalizations in the dictionary sense[3] can also be false.

10.1 Rationalizing Commonsense Auto-generated Explanations (CAGE)

Rajani et al. [92] provide explanations to the **Common sense Question Answering** (**CQA**) dataset, which is a multiple-choice question answering dataset [115]. The explanations are independent of the model and are provided via Amazon Mechanical Turk. To provide rationalization explanations, they then fine-tune a GPT model [89], using the question, answers, and explanation. See Equation (13) for an example of the exact prompt construction. To clarify, this GPT model is not the explained model but provides the explanations, this is known as an explainer-model.

$$\begin{aligned} \mathit{input} = \text{“}&\underbrace{\text{What could people do that involves talking?}}_{\text{question}}\ \underbrace{\text{confession}}_{\text{choice 1}}\ \text{,}\ \underbrace{\text{carnival}}_{\text{choice 2}} \\ &\text{, or}\ \underbrace{\text{state park}}_{\text{choice 3}}\ \text{?}\ \underbrace{\text{confession}}_{\text{answer}}\ \texttt{because}\ \text{”} \\ \mathit{target} = \text{“}&\underbrace{\text{confession is the only vocal action.}}_{\text{rational explanation}}\text{”} \end{aligned} \tag{13}$$

For simpler tasks, such as “Stanford Sentiment Treebank” [110], the prompt could simply be “`[input]. [answer] because [explanation]`”, see Figure 8 for hypothetical explanations using such a setup. Because *CAGE* uses a generative model, where `[answer]` can be a sequence, it is not limited to sequence-to-class problems.

[3]“the action of attempting to explain or justify behavior or an attitude with logical reasons, even if these are not appropriate.”—Oxford Defintion of *rationalization*.

	x	$p(y\|\mathbf{x};\theta)$	y
x	the year 's best and most unpredictable comedy	0.91	pos
	unpredictable comedies are funny	-	-
x	we never feel anything for these characters	0.95	neg
	it is important to feel for characters	-	-

Fig. 8. Hypothetical explanations from using *CAGE* to produce rationalizations for the prediction.

They find that rationalization explanations provide nearly identical explanations as reasoning explanations (those where the answer is not known by the explanation model). The method is validated to be *human-grounded*, by tasking humans to use the explanation to predict the model behavior, again they find identical performance.

It is questionable if *CAGE* is *functionally-grounded*, as its only connection to the explained model is during inference, where the `answer` is produced by the explained model. Because there are no other connections to the explained model, there is little reason to think the GPT explainer-model can reflect the models behavior. If the humans who provided explanations had specialist insight into the model, then an argument could be made for *CAGE* to be *functionally-grounded*. However, as the humans were Mechanical Turk workers, this is unlikely.

10.2 Discussion

Groundedness. This sub-field of natural *natural language* explanations has received criticism in NLP for not evaluating *functionally-grounded* [48]. This issue is even more problematic because the annotated explanations are provided by humans who have no insights into the model's behavior [128]. The explanation model therefore just learns about humans' thought processes rather than the model's logical process. This issue is somewhat unique to the NLP literature and is better treated in other fields [6].

Future work. Most work on natural *natural language* explanations uses *intrinsic* methods, under the motivation that forcing the model to "reason about itself" will make it more accurate. Unfortunately, this hypothesis has received criticism because the little *post-hoc* work there exist, show that this is not the case. Additionally, there are theoretical arguments for why this would not be the case [55].

11 CONCEPTS

A *concept explanation* attempts to explain the model, in terms of an abstraction of the input, called a *concept*. A classical example in computer vision, is to explain how the concept of stripes affects the classification of a zebra. Understanding this relationship is important, as a computer vision model could classify a zebra based on a horse-like shape and a Savana background. Such relation may yield a high accuracy score but is logically wrong.

The term *concept* is much more common in computer vision [44, 59, 80] than in NLP. Instead, the subject is often framed more concretely as bias-detection, in NLP. For example, Vig et al. [122] uses the concept of occupation-words like *nurse*, and relates it to the classification of the words *he* and *she*.

Regardless of the field, in both NLP and CV, only a single class or small subset of classes are analyzed. For this reason, *concept explanation* belong in its own category of *class explanations*. However, in the future, we will likely see more types of *class explanations*.

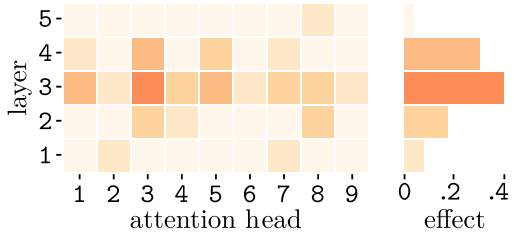

Fig. 9. Visualization of hypothetical NIE results, similar to Vig et al. [122]. Such visualization can reveal which attention-head are responsible for gender bias, in a small GPT-2 model. A stronger color indicates a higher NIE, meaning more responsible for the bias.

11.1 Natural Indirect Effect (NIE)

Consider a language model with the prompt $\mathbf{x}$ = "The nurse said that". To measure if the gender-stereotype of "nurse" is female, it is natural to compare $p(\text{she}|\mathbf{x};\theta)$ with $p(\text{he}|\mathbf{x};\theta)$, or alternatively $p(\text{they}|\mathbf{x};\theta)$. Generalized, Vig et al. [122] express this as

$$\text{bias-effect}(\mathbf{x};\theta) = \frac{p(\textit{anti-stereotypical}|\mathbf{x};\theta)}{p(\textit{stereotypical}|\mathbf{x};\theta)}\,. \tag{14}$$

Vig et al. [122] then provide insight into which parts of the model are responsible for the bias. They do this by measuring the ***Natural Indirect Effect*** (**NIE**) from causal mediation analysis. Although this approach applies to a sequence-to-sequence model, only one token being considered at a time. It is therefore possible also apply it to purely sequence-to-class models.

Given a model $f(\mathbf{x};\theta)$, mediation analysis is used to understand how a latent representation $z(\mathbf{x};\theta)$ (called the mediator) affects the final model output. This latent representation can either be a single neuron or several neurons, like an attention head. The *Natural Indirect Effect* measures the effect that goes through this mediator.

To measure causality, an *intervention* on the concept measured must be made. As intervention, Vig et al. [122] replace "nurse" with "man", or "woman" for oppositely biased occupations. They call this replace operation `set-gender`.

Then to measure the effect of the mediator Vig et al. [122] introduce

$$\text{mediation-effect}_{m_1,z,m_2}(\mathbf{x};\theta) = \frac{\text{bias-effect}_{z(m_2(\mathbf{x});\theta)}(m_1(\mathbf{x});\theta)}{\text{bias-effect}(\mathbf{x};\theta)}, \tag{15}$$

where $m \in \{\texttt{identity}, \texttt{set-gender}\}$ and $\text{bias-effect}_{z(m_2(\mathbf{x}))}(\cdot)$ is $\text{bias-effect}(\cdot)$ but uses a modified model with the mediator values for $z(m_1(\mathbf{x}))$ fixed. With this definition, the *Natural Indirect Effect* follows from causal mediation analysis literature [85].

$$\begin{aligned}\text{NIE}_z = \mathbb{E}_{\mathbf{x}\in\mathcal{D}}[&\,\text{mediation-effect}_{\texttt{identity},z,\texttt{set-gender}}(\mathbf{x};\theta)\\ &- \text{mediation-effect}_{\texttt{identity},z,\texttt{identity}}(\mathbf{x};\theta)].\end{aligned} \tag{16}$$

Vig et al. [122] apply *Natural Indirect Effect* to a small GPT-2 model, where the mediator is an attention head. By doing this, Vig et al. [122] can identify which attention heads are most responsible for the gender bias, when considering the occupation concept. Hypothetical results, but results similar to those presented in Vig et al. [122], are presented in Figure 9.

11.2 Discussion

Groundedness. As a new field, there is not much work on *groundedness*. Vig et al. [122] do not measure either *functionally-groundedness* or *human-groundedness* on *Natural Indirect Effect*. It is

also not obvious how *functionally-groundedness* could be measured. Note, that this situation is not unique to *concept explanation*, as many other communication approaches also do not have an established measure of *functionally-groundedness*.

Future work. Concept explanation requires either a new dataset or an annotation of an existing dataset. This can be quite expensive and impractical, especially when there is no concrete concept in mind and the user wants a more exploratory explanation. However, there is new research towards discovering concepts automatically [43].

12 VOCABULARY

For this category, we define the term *vocabulary explanation* as methods, which explain the whole model in relation to each word in the vocabulary and is therefore a *global explanation*.

In the sentiment classification context, a useful insight could be if positive and negative words are clustered together, respectively. Furthermore, perhaps there are words in those clusters which can not be considered of either positive or negative sentiment. Such a finding could indicate a bias in the dataset.

Because *vocabulary explanations* explain using the model's vocabulary, they can often be applied to both sequence-to-class and sequence-to-sequence models. This is especially true for explanations based on the embedding matrix, which so is almost exclusively the case.

Because an embedding matrix is often used and because neural NLP models often use pre-trained word embeddings, most research on *vocabulary explanations* is applied to the pre-trained word embeddings [77, 87]. However, in general, these explanation methods can also be applied to the word embeddings after training.

12.1 Projection

A common visual explanation is to project embeddings to two or three dimensions. This is particularly attractive, as word embeddings are of a fixed number of dimensions, and can therefore draw from the very rich literature on projection visualizations of tabular data, most notable is perhaps **Principal Component Analysis** (**PCA**) [86].

t-SNE. Another popular and more recent method is t-SNE [120], which has been applied to word embeddings [66]. This method has in particular been attractive as it allows for non-linear transformations, while still keeping points that are close in the word embedding space, also close in the visualization space. t-SNE does this by representing the two spaces with two distance-distributions, it then minimizes the KL-divergence by moving the points in the visualization space.

Note that Li et al. [66] does not go further to validate t-SNE in the context of word embeddings, except to highlight that words of similar semantic meaning are close together, we provide a similar example in Figure 10.

Supervised projection. A problem with using PCA and t-SNE, is that they are unsupervised. Hence, while they might find a projection that offers high contrast, this projection might not correlate with what is of interest. An attractive alternative is therefore to define the projection, such that it reveals the subject of interest.

Bolukbasi et al. [19] are interested in how gender-biased a word is. They explore gender-bias, by projecting each word onto a gender-specific vector and a gender-neutral vector. Such vectors can either be defined as the directional vector between "he" and "she", or alternative. Bolukbasi et al. [19] also use multiple gender-specific pairs such as "daughter-son" and "herself-himself", and then use their first Principal Component as a common projection vector.

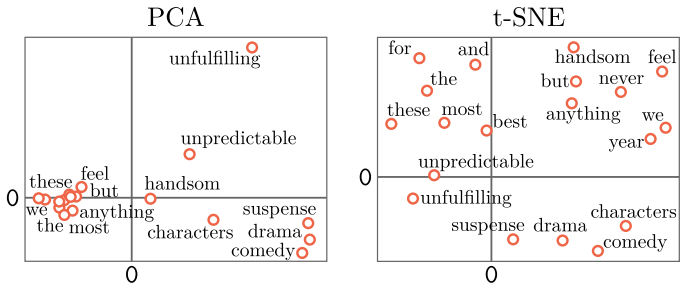

Fig. 10. PCA [86] and t-SNE [120] projection of GloVe [87] embeddings for the words in the semantic classification examples, as shown in Section 3 and elsewhere in this survey.

Table 2. Fictive Example of the Top-3 Words for Each Basis-dimension in the Rotated Word Embeddings

Basis-dimension	top-3 words
1	handsome, feel, unpredictable
2	most, best, anything
3	suspense, drama, comedy

12.2 Rotation

The category of, for example, all positive sentiment words may have similar word embeddings. However, it is unlikely that a particular basis dimension describes positive sentiment itself. A useful interpretability method, is therefore to rotate the embedding space such that the basis-dimensions in the new rotated embedding space represents significant concepts. This is distinct from *projection* methods because there is no loss of information as only a rotation is applied.

Park et al. [84] perform such rotation using ***Exploratory Factor Analysis (EFA)*** [33]. The idea is to formalize a class of rotation matrices, called the *Crawford-Ferguson Rotation Family* [34]. The parameters of this rotation formulation are then optimized, to make the rotated embedding matrix only have a few large values in each row or column. As an hypothetical example see Table 2.

Park et al. [84] validate this method to be *human-grounded* by using the *word intrusion* test. The classical word Intrusion test [26] provides six words to a human annotator, five of which should be semantically related, and the 6th is the intruder which is semantically different. The human annotator then has to identify the intruder word. Importantly, semantic relatedness is in this case defined as the top-5 words of a given basis-dimension in the rotated embedding matrix.

Unfortunately, rather than having humans detect the intruder, Park et al. [84] use a distance ratio, related to the cosine-distance, as the detector. This is problematic, as distance is directly related to how the semantically related words were chosen. In this case, the intruder should have been identified either by a human or an oracle model.

12.3 Discussion

Groundedness. In terms of *human-grounded*, *vocabulary explanation* are one of the few sub-fields that have a well-established test, namely the *word intrusion* test [26]. It is therefore hard to justify when methods in this category replace humans with an algorithm, as this largely invalidates the test.

Future work. While past work, such as **Latent Dirichlet Allocation** (**LDA**) [18], has provided great *vocabulary explanations*, contemporary work using neural networks is quite limited and is

mostly based on the embedding matrix. This is a pity, as the embedding matrix only provides a limited picture and it is not hard to imagine using other information sources to create *vocabulary explanations*. For example, one could aggregate the word-contributions provided by *input feature* explanations.

13 ENSEMBLE

Ensemble explanations attempt to provide a *global explanation* by collecting multiple *local explanations*. This is done such that each *local explanation* represents the different modes of the model.

The extreme of this idea would be to provide a *local explanation* for every possible input, thereby providing a *global explanation*. Unfortunately, such an explanation is too much information for a human to understand and would not be *human-grounded*. As Miller [78] state, an explanation should be selective. The task of *ensemble* explanations, is therefore to strategically select representative examples and their corresponding *local explanations*.

The assumption is that the model operates within different modes. Furthermore, that one example, or a few examples, from each mode can sufficiently represent the models entire behavior. For example, in sentimate classification of movie reviews, a model may have one behavior for comments about the acting, another behavior for comments about the music score, and so on.

Ensemble explanations are a very broad category of explanations, as for every type of *local explanation* method there is, an *ensemble* explanation could in principle be constructed. As such, if it can be applied to sequence-to-class or sequence-to-sequence models depends on the specific method. However, in practice, very few *ensemble* methods have been proposed, and most of them apply only to tabular data [52, 93, 102].

13.1 Submodular Pick LIME (SP-LIME)

SP-LIME by Ribeiro et al. [94] attempts to select B observations (a budget), such that they represent the most important features based on their *LIME* explanation, an example of this can be seen in Figure 11. Note that, while *LIME* explanations can be made for each output token and can therefore be used in a sequence-to-sequence context, *SP-LIME* do assume a sequence-to-class model.

SP-LIME calculates the importance of each feature v, by summing the absolute importance for all observations in the dataset, this total importance is $\mathbf{I}_v$ in Equation (17). The objective is then to maximize the sum of $\mathbf{I}_v$ given a subset of features, by strategically selecting B observations. Note that selecting multiple observations which represent the same features will not improve the objective. The specific objective is formalized in Equation (17), which Ribeiro et al. [94] optimize greedily.

$$\begin{aligned}\mathbf{G}_{\text{SP-LIME}} = \underset{\tilde{\mathcal{D}} \text{ s.t.} |\tilde{\mathcal{D}}| \leq B}{\operatorname{argmax}} \sum_{v=1}^{V} \mathbb{1}_{\left[\exists \tilde{\mathbf{x}}_i \in \tilde{\mathcal{D}} : \left|\mathbf{E}_{\text{LIME}}\left(\tilde{\mathbf{x}}_i, \operatorname{argmax}_i p(i|\tilde{\mathbf{x}}_i;\theta)\right)_v\right| > 0\right]} \mathbf{I}_v \\ \text{where } \tilde{\mathcal{D}} \subseteq \mathcal{D} \\ \mathbf{I}_v = \sum_{\tilde{\mathbf{x}}_i \in \mathcal{D}} \left|\mathbf{E}_{\text{LIME}}\left(\tilde{\mathbf{x}}_i, \underset{i}{\operatorname{argmax}}\, p(i|\tilde{\mathbf{x}}_i;\theta)\right)_v\right|.\end{aligned} \tag{17}$$

A major challenge with *SP-LIME* is that it requires computing a *LIME* explanation for every observation. Because each *LIME* explanation involves optimizing a logistic regression this can be quite expensive. To reduce the number of observations that need to be explained, Sangroya et al. [102] proposed using *Formal Concept Analysis* to strategically select which observations to explain. However, this approach has not yet been applied to NLP.

Ribeiro et al. [94] validate *SP-LIME* to be *human-grounded* by asking humans to select the best classifier, where a "wrong classifier" is trained on a biased dataset and a "correct classifier" is

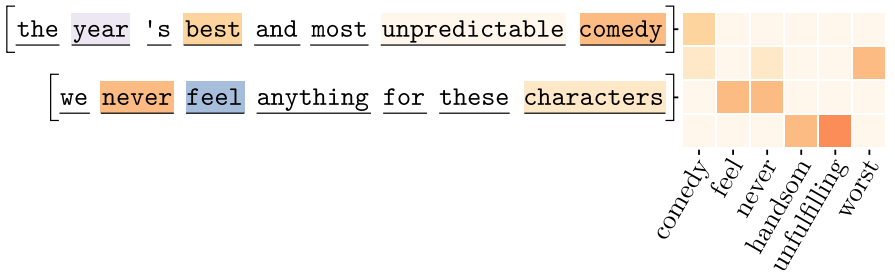

Fig. 11. Visualization of *SP-LIME* in a hypothetical setting. The matrix shows how each selected observation represents the different modes of the model. The left-side shows two out of the four selected example and their *LIME* explanation.

trained on a curated dataset. Ribeiro et al. [94] then compare *SP-LIME* with a random baseline, which simply selects random observations. From this experiment, they find that 89% of humans can select the best classifier using *SP-LIME*, whereas only 75% can select the best classifier based on the random baseline.

13.2 Discussion

Groundedness. The *functionally-groundedness* of *ensemble* explanations is very much dependent on the *functionally-groundedness* of the *local explanation*. It is therefore diffcult to imagine a general evaluation approach for *ensemble* explanations. However, even for *local explanations* with established validation *functionally-groundedness* does not come for free, as also the selection algorithm also needs to be validated.

Future work. As mentioned there is not much work using *ensemble* explanations. This is because when non-tabular data is used, it is more challenging to compare the selected explanations to ensure they represent different modes. Even *SP-LIME* [94] which does apply to NLP tasks, uses a BoW representation as a tabular proxy. Additionally, we can imagine that *ensemble* explanations are hard to scale, as datasets increases and models get more complex with more modes.

That being said, we would be curious to see more work in this category. For example, an *ensemble* explanation which used a *influential example* method to show the overall most relevant observations.

14 LINGUISTIC INFORMATION

To validate that a natural language model does something reasonable, a popular approach is to attempt to align the model with the large body of linguistic theory that has been developed for hundreds of years. Because these methods summarize the model, they are a case of a *global explanation*.

Methods in this category either probe by strategically modifying the input to observe the model's reaction or show alignment between a latent representation and some linguistic representation. The former is called *behavioral probes* or *behavioral analysis*, the latter is called *structural probes* or *structural analysis*. Which type of models these strategies applies to depends on the specific method. However, in general, *behavioral probes* applies primarily to sequence-to-class models and *structural probes* applies to both sequence-to-class and sequence-to-sequence models.

One especially noteworthy subcategory of *Structural Probes* is *BERTology*, which specifically focuses on explaining the BERT-like models [20, 36, 70]. BERT's popularity and effectiveness have resulted in countless articles in this category [28, 30, 76, 98, 116], hence the name *BERTology*. Some

of the works use the attention of BERT and are therefore *intrinsic* explanations, while others simply probe the intermediate representations and are therefore *post-hoc* explanations.

There already exist well-written survey articles on *Linguistic Information* explanations. In particular, Belinkov et al. [14] cover *behavioral probes* and *structural probes*, Rogers et al. [98] discuss *BERTology*, and Belinkov and Glass [15] cover *structural probing* in detail. In this section, we will therefore not go in-depth, but simply provide enough context to understand the field and importantly mention some of the criticisms, that we believe have not been sufficiently highlighted by other surveys.

14.1 Behavioral Probes

The research being done in *behavioral probes*, also called *behavioral analysis*, is not just for interpretability but also to measure the robustness and generalization ability of the model. For this reason, many *challenge datasets* are in the category of *behavioral analysis*. These datasets are meant to test the model's generalization capabilities, often by containing many observations of underrepresented modes in the training datasets. However, the model's performance on *challenge datasets* does not necessarily provide interpretability.

One of the initial articles providing interpretability via *behavioral probes* is that by Linzen et al. [67]. They probe a language model's ability to reason about subject-verb agreement correctly. A recent work, by Clouatre et al. [29], Sinha et al. [107], find that destroying syntax by shuffling words does not significantly affect a model trained on an NLI task, indicating that the model does not achieve natural language understanding.

As mentioned, this area of research is quite large and Belinkov et al. [14] cover *behavioral probes* in detail. Therefore, we just briefly discuss the work by McCoy et al. [74], which provides a particularly useful example on how *behavioral probes* can be used to provide interpretability.

McCoy et al. [74] look at **Natural Language Inference** (**NLI**), a task where a premise (for example, "The judge was paid by the actor") and a hypothesis (for example, "The actor paid the judge") are provided, and the model should inform if these sentences are in agreement (called *entailment*). The other options are *contradiction* and *neutral*. McCoy et al. [74] hypothesise that models may not actually learn to understand the sentences but merely use heuristics to identify *entailment*.

They propose three heuristics based on the linguistic properties: lexical overlap, subsequence, and constituent. An example of lexical overlap is the premise "**The doctor** was **paid** by **the actor**" and hypothesis "The doctor paid the actor". The proposed heuristic is that this observation would be classified as *entailment* by the model due to lexical overlap, even though this is not the correct classification.

To test for these heuristics, McCoy et al. [74] developed a dataset, called HANS, which contains examples with these linguistic properties but do not have *entailment*. The results (Table 3) validates the hypothesis that the model relies on these heuristics rather than a true understanding of the content. Had just an average score across all heuristics been provided, this would just be a robustness measure. However, by providing meta-information on which pattern each observation follows, the accuracy scores provide interpretability on where the model fails.

In terms of *functionally-groundedness*, McCoy et al. [74] perform no explicit evaluation. However, given that *behavioral probes* merely evaluate the model, *functionally-groundedness* is generally not a concern. Furthermore, while McCoy et al. [74] do evaluate with humans, this is not a *human-grounded* evaluation. Because they only use humans to evaluate the dataset, not if the explanation itself is suitable to humans.

Table 3. Performance on the HANS Dataset Provided by McCoy et al. [74]

	Lexical Overlap	Subsequence	Constituent	Average
BERT [36]	17%	5%	17%	–
Human (Mechanical Turk)	–	–	–	77%

Unfortunately, McCoy et al. [74] do not provide enough information to make a direct comparison possible. For comparison, BERT has 83% accuracy on MNLI [131], which was used for training.

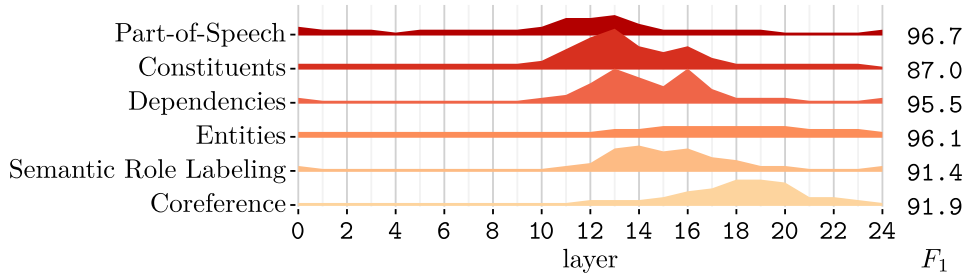

Fig. 12. Results by Tenney et al. [116] which shows how much each BERT [36] layer is used for each linguistic task. The F_1 score for each task is also presented.

14.2 Structural Probes

Probing methods primarily use a simple neural network, often just a logistic regression, to learn a mapping from an intermediate representation to a linguistic representation, such as the **Part-Of-Speech** (**POS**).

One of the early articles, by Shi et al. [106], analyzed the sentence-embeddings of a sequence-to-sequence LSTM, by looking at POS, TSS, SPC (the smallest phrase constituent for each word), tense (past or non-past), and voice (active or passive). Similarly, Adi et al. [4] used a **multi-layer-perceptron** (**MLP**) to analyze sentence-embeddings for sentence-length, word-presence, and word-order. More recently Conneau et al. [31] have been using similar linguistic tasks and MLP probes but have extended previous analyses to multiple models and training methods.

Analog to these articles, a few methods use cluster algorithms instead of logistic regression [22]. Additionally, some methods only look at *word embeddings* [62]. The list of articles is very long, we suggest looking at the survey article by Belinkov and Glass [15].

BERTology. As an instructive example of probing in BERTology, the article by Tenney et al. [116] is briefly described. Note that this is just one example of a vast number of articles. Rogers et al. [98] offer a much more comprehensive survey on BERTology.

Tenney et al. [116] probe a BERT model [36] by computing a learned weighted-sum $\mathbf{z}_i(\mathbf{x};\theta)$ for each intermediate representation $\mathbf{h}_{l,i}(\mathbf{x};\theta)$ of the token i, as described in Equation (18).

$$\mathbf{z}_i(\mathbf{x};\theta) = \gamma \sum_{l=1}^{L} s_l \mathbf{h}_{l,i}(\mathbf{x};\theta) \quad \text{where } \mathbf{s} = \text{softmax}(\mathbf{w}). \tag{18}$$

The weighted-sum $\mathbf{z}_i(\mathbf{x})$ is then used by a classifier [118], and the weights s_l, parameterized by $\mathbf{w}$, describe how important each layer l is. The results can be seen in Figure 12.

Criticisms. A growing concern in the field of probing methods is that given a sufficiently high-dimensional embedding, complex probe, and large auxiliary dataset, the probe can learn everything from anything. If this concern is valid, it would mean that the probing methods do not provide *functionally-grounded explanations* [13].

Recent work attempts to overcome this concern by developing baselines. Zhang and Bowman [137] suggest learning a probe from an untrained model, as a baseline. In that article, they find probes can indeed achieve high accuracy from an untrained model unless the auxiliary dataset size is decreased dramatically. Similarly, Hewitt and Liang [49] use randomized datasets as a baseline, called a control task. For example, for POS they assign a random POS-tag to each word, following the same empirical distribution of the non-randomized dataset. They find that equally high accuracy can be achieved on the randomized dataset unless the probe is made extraordinarily small.

Information-Theoretic Probing. The solutions presented by Zhang and Bowman [137] and Hewitt and Liang [49] are useful. However, limiting the probe and dataset size could make it impossible to find complex hidden structures in the embeddings.

Voita and Titov [123] attempt to overcome the criticism by a more principled approach, using information theory. More specifically, they measure the required complexity of the probe as a communication effort, called ***Minimum Description Length*** (**MDL**), and compare the MDL with a control task similar to Hewitt and Liang [49]. They find, similar to Hewitt and Liang [49], that the probes achieve similar accuracy on the probe dataset as on the control task. However, the control task is much harder to communicate (the MDL is higher), indicating that the probe is much more complex, compared to training on the probe dataset.

14.3 Discussion

Groundedness. Considering the vast amount of research on *linguistic information* explanations, we find it worrying that there is not more work on evaluating if these explanations are actually useful, in terms of the *human-groundedness* and *functionally-groundedness*. Without such evaluation, it is difficult to ensure that the field of *linguistic information* explanations moves in a productive direction.

Future work. Considering the *groundedness* issues in *linguistic information* explanations, we advocate for more focus on *groundedness*. Voita and Titov [123] provide a great solution to how the *functionally-groundedness* issues can be overcome. However, the field still lacks independent study on *human-groundedness* and *functionally-groundedness*.

15 RULES

Rule explanations attempt to explain the model by a simple set of rules, therefore they are an example of *global explanations*.

Reducing highly complex models like neural networks to a simple set of rules is likely impossible. Therefore, methods that attempt this simplify the objectivity by only explaining one particular aspect of the model.

Due to the challenges of producing rules, there is little research attempting it. We will present *Compositional Explanations of Neurons* [80] and *SEAR* [96].

15.1 Semantically Equivalent Adversaries Rules (SEAR)

SEAR is an extension of the ***Semantically Equivalent Adversaries*** (**SEA**) method [96], where they developed a sampling algorithm for finding adversarial examples. Hence, the rule-generation objective is simplified, as only rules that describe what breaks the model needs to be generated.

		$p(y\|\mathbf{x};\theta)$	y	Flips
$\mathbf{x}$	the year 's best and most unpredictable comedy	0.91	pos	-
$\tilde{\mathbf{x}}$	the best and most unpredictable comedy this year	0.13	-	-
rule	DET year 's → this year	-	-	1%
$\mathbf{x}$	we never feel anything for these characters	0.95	neg	-
$\tilde{\mathbf{x}}$	we never empathize for these characters	0.11	-	-
rule	feel → empatize	-	-	4%

Fig. 13. Hypothetical example showing rules which commonly break the model. The flip-rate describes how often these rules break the model. $\mathbf{x}$ represents the original sentence and $\tilde{\mathbf{x}}$ represents an adversarial example.

Additionally, because *SEAR* uses an *adversarial examples* explanation, it only applies to sequence-to-class models.

Ribeiro et al. [96] propose rules by simply observing individual word changes found by the *SEA* method discussed earlier, and then compute statistics on the bi-grams of the changed word and the Part of Speech of the adjacent word, Figure 13 shows examples of this. If the proposed rule has a high success-rate (called filp-rate), in terms of providing a semantically equivalent adversarial sample, it is considered a rule.

The authors validate this approach by asking experts to produce rules, and then compare the success-rate of human-generated rules and *SEAR*-generated rules. They find that the rules generated by *SEAR* have a higher success-rate.

15.2 Discussion

Future work. As mentioned, there is little work on *rule* explanations. While this is definetly due to the inherent challenge, it is not too hard to imagine something like the *Anchor* method be modified towards *global explanation*, in which case it would be a *rule* explanation.

Groundedness. Because the category of *rule* explanations can be very diverse, *groundedness* evaluation would likely depend on the specific explanation method. However, generally, *functionally-groundedness* can be measured by asserting if the rule holds true by evaluating it on the dataset and compare with the model response. Additionally, *human-groundedness* can be evaluated by asking humans to predict the model's output or choose the better model.

16 LIMITATIONS

While it is the goal of this survey to provide an overview and categorization of current post-hoc interpretability for neural NLP models, we also recognize that the field is too vast to include all works in this survey. To decide what works to include, the overall has been to focus on diversity in terms of communication approach and information used. Essentially, to make Table 1 as comprehensive as possible.

Communication approaches like *input features* and *linguistic information* have a particularly large amount of literature, which we did not discuss, as that would outweigh other communication approaches. For these two approaches, we focus on highlighting the progression of the field.

Beyond this overarching limitation, the following two limitations are worth discussing.

Quantitative comparisons. Ideally, this survey would include quantitative comparisons of the methods. However, there currently does not exist an unified and principled benchmark yet.

Producing a principal benchmark is in itself extreamely difficult and out of scope for this survey, in Section 18 we discuss further where this difficulty comes from. Performing quantitative comparisons would therefore best be left for future work on interpretability benchmarks.

Visual examples. Because communication is essential to this survey, visual examples of how the method communicates have been provided throughout this survey. These examples are however fictive and optimistic, showing often the best case for each explanation method. However, in practice, accurate and highly useful explanations can only be produced for some examples for *local explanations*, or some datasets in the case of *class and global explanations*. Furthermore, the visualizations are not necessarily the most effective visualizations but are instead what we believe to be the most canonical visualizations.

How an explanation method should be visualized is its own field of study and should draw from human-computer interface literature. This is something that was not covered in this survey.

17 FINDINGS

This survey covers a large range of methods. In particular, we discuss how each method communicates and is evaluated. However, some discussion is not specific to any motivation, measure, or method for interpretability. Therefore, this section covers a few valuable findings which should be discussed from a holistic perspective.

Terminology. Because interpretability is an emerging field, terminology still varies significantly from article to article. In particular, the terminology regarding measures of interpretability vary. For example, *human-groundedness* is often confused with *functionally-groundedness*, and for each measure category, there are synonyms such as *simulatability*, and *comprehensibility* for *human-groundedness*. Additionally, the terms for the communication types are sometimes confused. Especially, *adversarial examples,* and *counterfactuals* are occasionally interchanged.

This survey does not seek to unify the terminology, but we hope it will at least serve as a source to understand which terms mean the same and which terms are different.

Synergy. Methods from different communication approaches can benefit each other. For example, both the *adversarial examples* method *HotFlip* and the *counterfactual* method *MiCE* uses the *gradient w.r.t. the input* method from the *input feature* explanation literature. Recognizing these connections allows for flexibility in explanation methods. In the aforementioned example, other *input feature* explanations could have been used as well. Additionally, criticisms on the faithfulness of *input feature* methods could affect its dependents.

Helpful complex models. Models like GPT and T5 are immensely complex and thereby contribute to the interpretability challenge. However, importantly these models are not exclusively bad from an interpretability perspective, as they are also used to provide fluent explanations. For example, in *counterfactual* explanations, *Polyjuice* uses the GPT-2 model and *MiCE* uses the T5 model. Similarly, in *natural language* explanations, *CAGE* uses GPT. As such, these complex models can not be said to be exclusively counterproductive to interpretability.

18 FUTURE DIRECTIONS AND CHALLENGES

Interpretability for NLP is a fast-growing research field, with many methods being proposed each year. This survey provides an overview and categorization of many of these methods. In particular, we present Table 1 as a way to frame existing research. It is also the hope that Table 1 will help frame future research. In this section, we provide our opinions on what the most relevant challenges and future directions are in interpretability.

Measuring Interpretability. How interpretability is measured varies significantly. Throughout this article, we have briefly documented how each method measures interpretability. A general observation is that each method article often introduces its own measures of *functionally-groundedness* or *human-groundedness*. Even when established standards exist, such as the *word intrusion test* [26], they get modified. This trend reduces comparability and risks invalidating the measure itself.

It is important to recognize that measuring interpretability is, in some cases, inherently difficult. For example, in the case of measuring the *functionally-groundedness* of *input feature* explanations, it is inherently impossible to provide gold labels for what is a correct explanation, because if humans could provide gold labels we would not need the explanation in the first place. This fundamentally leaves only proxy measures and axioms of *functionally-groundedness*. However, this does not mean highly principled proxy measures can not be developed [51].

For this reason, we are encouraging researchers and reviewers to value principled articles on measuring interpretability. Even if those measures do not become established standards, a dedicated focus on measuring interpretability is a necessity for the integrity of the interpretability field.

Class explanations. There is a large number of articles on explanation methods. However, *class explanations* remain an underrepresented middle ground between *local* and *global explanations*.

The specific communication approach chosen should reflect its application, and for this reason, no explanation type can be said to be superior. However, it's important to recognize that *local explanations* can only provide anecdotal evidence and *global explanations* can be too abstract to ground what is explained. As such *class explanations* have their value, as they are not specific enough to be anecdotal. Simultaneously, they are grounded in the class they explain, making them easier to reason about. For this reason, we would encourage that *class explanation* to gain equal representation in interpretability research.

Sequence-to-sequence explanations. In this survey, we frequently comment on if a explanation method can be applied to sequence-to-class models or sequence-to-sequence models. Most methods are primarily made for sequence-to-class models, and the few that apply to sequence-to-sequence models are often not directly made for that purpose.

We suspect a reason for primarily explaining sequence-to-class models, is that sequence-to-sequence explanations may depend on interactive visualization to a greater extent [72, 111, 117], which is harder to implement and write about in typical machine learning venues.

Regardless, sequence-to-sequence models are widely used in real-life applications, for example in machine translation. We therefore advocate for developing more explanations for sequence-to-sequence models, or at the very least include an evaluation on a sequence-to-sequence model in articles that provide methods that can operate on both types of models.

Combining post-hoc with intrinsic methods. Post-hoc and *intrinsic* methods are in literature, including this article, represented as distinct. However, there are important middle grounds.

As mentioned in the introduction, most *intrinsic* methods are not purely intrinsic. They often have an intermediate representation, which can be intrinsically interpretable. However, producing this representation is often done with a black-box model. For this reason, *post-hoc* explanations are needed if the entire model is to be understood.

Beyond this direction, there are works where the training objective and procedure helps to provide better *post-hoc* explanations. This survey briefly argues that the *Kernel SHAP* method exists in this middle ground, as it depends on input-masking being part of the training procedure. In

computer vision, Bansal et al. [10] show that adding noise to the input images creates better *input feature* explanations. In general, we hope to see more work in this direction.

19 CONCLUSION

This survey presents an overview of *post-hoc* interpretability methods for neural networks in NLP. The main content of this survey is on the interpretability methods themself and how they communicate their explanation of the model. This content is categorized through Table 1.

Throughout the survey, we also refer back to *measures of interpretability* (Section 4) to describe how each article evaluates its proposed method. Measuring interpretability is an often undervalued aspect of interpretability with little standardization of the benchmarks. However, by briefly mentioning each method of measurement, we hope that this will lead to less fragmentation.

Finally, we discuss interesting findings and future directions, which we consider particularly important. Overall, we hope that Table 1, the discussions of each communication approach and their methods, and the final discussion sections help frame future research and provide broad insight to those who apply interpretability.

APPENDICES

A INPUT FEATURES

A.1 Integrated Gradient (IG)

The *gradient* approach has been further developed, the most notable development is *Integrated Gradient* [114].

Sundararajan et al. [114] primarily motivate *Integrated Gradient* via the desirables they call *sensitivity* and *completeness*. *Sensitivity* means, if there exists a combination of $\mathbf{x}$ and baseline $\mathbf{b}$ (often an empty sequence), where the logit outputs of $f(\mathbf{x};\theta)$ and $f(\mathbf{b};\theta)$ are different, then the feature that changed should get a non-zero attribution. This desirable is not satisfied for the gradient method, for example, due to the truncation in ReLU$(\cdot)$. *Completeness* means, the sum of importance scores assigned to each token should equal the model output relative to the baseline $\mathbf{b}$.

To satisfy these desirables, Sundararajan et al. [114] develop Equation (19) which integrates the gradients between an uninformative baseline $\mathbf{b}$ and the observation $\mathbf{x}$ [114].

$$\mathbf{E}_{\text{integrated-gradient}}(\mathbf{x}, c) = (\mathbf{x} - \mathbf{b}) \odot \frac{1}{k} \sum_{i=1}^{k} \nabla_{\tilde{\mathbf{x}}_i} f(\tilde{\mathbf{x}}_i;\theta)_c, \quad \tilde{\mathbf{x}}_i = \mathbf{b} + {}^{i}/_{k}(\mathbf{x} - \mathbf{b}), \tag{19}$$

where $f(\mathbf{x};\theta)$ is the model logits.

This approach has been successfully applied to NLP, where the uninformative baseline can be an empty sentence, such as padding tokens [81].

Although Integrated Gradient has become a popular approach, it has recently received criticism in the **computer vision** (**CV**) community for not being *functionally-grounded* [51]. More recent work have applied a similar analysis to NLP, and found that the *functionally-groundedness* is at the very least task dependent [73]. Additionally, Bastings et al. [11] uses synthetic NLP tasks and arrived at the same task-dependent conclusion. One explanation for the lack of *functionally-groundedness* is the input multiplication which is not directly related to the model [3].

A.2 Kernel SHAP

A limitation of *LIME* is that the weights in a linear model are not necessarily *intrinsically* interpretable. When there exists multicollinearity (input features are linearly correlated with each other) then the model weights can be scaled arbitrarily creating a false sense of importance.

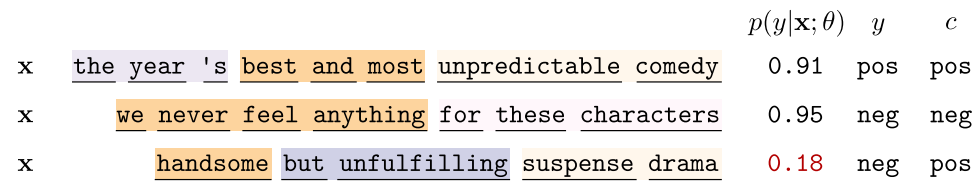

Fig. A.1. Fictive visualization of *Kernel SHAP*. Note how input tokens are combined to a single feature to make *SHAP* more tractable to compute, this is the role of $h_{\mathbf{x}}(z)$ in Equation (20).

To avoid the multicollinearity issue, one approach is to compute Shapley values [105] which are derived from game theory. The central idea is to fit a linear model for every permutation of features enabled. For example, if there are two features $\{x_1, x_2\}$, the Shapley values would aggregate the weights from fitting the datasets with features $\{\varnothing\}, \{x_1\}, \{x_2\}, \{x_1, x_2\}$. If there are T features this would require $O(2^T)$ models.

While this method works in theory, it is clearly intractable. Lundberg and Lee [71] present a framework for producing Shapley values in a more tractable manner. The model-agnostic approach they introduce is called *Kernel SHAP*. It combines three ideas: it reduces the number of features via a mapping function $h_{\mathbf{x}}(\mathbf{z})$ as seen in Figure A.1, it uses squared-loss instead of cross-entropy by working on logits, and it weighs each observation by how many features there are enabled.

$$\begin{aligned} \mathbf{E}_{\text{SHAP}}(\mathbf{x}, c) &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{\mathbf{z} \in \mathbb{Z}^M} \pi(\mathbf{z}) \left(f(h_{\mathbf{x}}(\mathbf{z}); \theta)_c - g(\mathbf{z})\right)^2 \\ &\text{where } g(\mathbf{z}) = \mathbf{w}\mathbf{z} \\ \pi(\mathbf{z}) &= \frac{M-1}{(M \text{ choose } |\mathbf{z}|)|\mathbf{z}|(M-|\mathbf{z}|)}. \end{aligned} \tag{20}$$

In Equation (20), $\mathbf{z}$ is a $\{0,1\}^M$ vector that describes which combined features are enabled. This is then used in $h_{\mathbf{x}}(\mathbf{z})$, which enables those features in $\mathbf{x}$. Furthermore, $\mathbb{Z}^M$ represents all permutations of enabled combined features and $|\mathbf{z}|$ is the number of enabled combined-features. Figure A.1, demonstrates a fictive example of how input features can be combined and visualize their Shapley values.

Lundberg and Lee [71] shows *functionally-groundedness* by using that Shapley values uniquely satisfy a set of desirables and that *SHAP* values are also Shapley values. Furthermore, Lundberg and Lee [71] show *human-groundedness* by asking humans to manually produce importance measures and correlate them with the *SHAP* values.

A criticism of both SHAP and LIME is that they depend on pertubation of the input, this makes it possible to create adversarial models that appear ethical when explained using pertubated inputs but is in reality not ethical when evaluated without pertubation [108]. This means that LIME and SHAP can only provide a *functionally-grounded* explanation as long as the model is trained without malicious intent.

SHAP and Shapley values in general are heavily used in the industry [17]. In NLP literature *SHAP* has been used by Wu et al. [134]. This popularity is likely due to their mathematical foundation and the `shap` library. In particular, the `shap` library also presents Partition SHAP which claims to reduce the number of model evaluations to M^2, instead of 2^M.[4] One major disadvantage of *SHAP* is it inherently depends on the masked inputs still being valid inputs. For some NLP models, this

[4]See documentation https://shap.readthedocs.io/en/latest/example_notebooks/tabular_examples/model_agnostic/Simple%20Boston%20Demo.html

		$p(y\|\mathbf{x};\theta)$	y	c
x	the year 's best and most unpredictable comedy	0.91	pos	pos
x	we never feel anything for these characters	0.95	neg	neg
x	handsome but unfulfilling suspense drama	0.18	neg	pos

Fig. A.2. Fictive visalization, showing the *anchors* that are responsible for the prediction.

can be accomplished with a [MASK] token, while for it is not possible in a *post-hoc* setting. For this reason, *SHAP* exists at an intersection between *post-hoc* and *intrinsic* interpretability methods. This intersection is discussed more in Section 18.

A.3 Anchors

A further development of the idea, that sparse explanations are easier to understand, is *Anchors*. Instead of giving an importance score, like in the case of the gradient-based methods or *LIME*, the *Anchors* simply provides a shortlist of words that were most relevant for making the prediction [95]. The authors show *human-groundedness* with a similar user setup as in *LIME* [94].

The list-of-words called "anchors" (A) is formalized in Equation (21). Note that $c = \text{argmax}_i\, p(i|\mathbf{x};\theta)$ is a requirement for *anchors*, as using $\text{prec}(A) = \mathbb{E}_{\mathcal{D}(\tilde{\mathbf{x}}|A)}[\mathbb{1}_{y=\tilde{y}}]$ in Equation (21) would cause *anchors* to be unaffected by the model.

$$
\begin{aligned}
\mathbf{E}_{\text{anchors}}(\mathbf{x}) = & \underset{A \text{ s.t. } \text{prec}(A)\geq\tau \wedge A(\mathbf{x})=1}{\text{argmax}} \quad \text{cov}(A) \\
& \text{where } \text{prec}(A) = \mathbb{E}_{\mathcal{D}(\tilde{\mathbf{x}}|A)}\left[\mathbb{1}_{[\text{argmax}_i\, p(i|\mathbf{x};\theta)=\text{argmax}_i\, p(i|\tilde{\mathbf{x}};\theta)]}\right] \\
& \qquad\quad \text{cov}(A) = \mathbb{E}_{\mathcal{D}(\tilde{\mathbf{x}})}\left[A(\tilde{\mathbf{x}})\right] \\
& \qquad\quad A(\mathbf{x}) = \begin{cases} 1 & \text{if the anchors } A \text{ are in } \mathbf{x} \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}
\tag{21}
$$

This formalization says the anchor words should have the highest coverage ($\text{cov}(A)$), meaning the most sentences in the dataset $D(\tilde{\mathbf{x}})$ contains the anchors A. Furthermore, only consider anchors A that are sufficiently precise ($\text{prec}(A) \geq \tau$) and in $\mathbf{x}$. Precision is defined as the ratio of observations $\tilde{\mathbf{x}}$ with anchors A, denoted $\mathcal{D}(\tilde{\mathbf{x}}|A)$, where the predicted label of $\tilde{\mathbf{x}}$ matches the predicted label of $\mathbf{x}$.

Solving this optimization problem exactly is infeasible, as the number of anchors is combinatorially large. To approximate it, Ribeiro et al. [95] model $\text{prec}(A) \geq \tau$ probabilistically [57] and then use a bottom-up approach, where they add a new word to the k-best anchor candidate in each iteration similar to beam-search.

B ADVERSARIAL EXAMPLES

B.1 Semantically Equivalent Adversaries (SEA)

An alternative approach to produce adversarial examples that are ensured to be paraphrases is to sample from a paraphrasing model $q(\tilde{\mathbf{x}}|\mathbf{x})$. Ribeiro et al. [96] do this by measuring a semantical-equivalency-score $S(\mathbf{x}, \tilde{\mathbf{x}})$, as the relative likelihood of $q(\tilde{\mathbf{x}}|\mathbf{x})$ compared to $q(\mathbf{x}|\mathbf{x})$. It is then possible to maximize the similarity, while still having a different model prediction, an example of this is seen in Figure B.1. The exact method is defined in Equation (22), which also constrains the optimization

	$\mathbf{x}$	$p(y\|\mathbf{x};\theta)$	y	$S(\mathbf{x},\tilde{\mathbf{x}})$
$\mathbf{x}$	the year 's best and most unpredictable comedy	0.91	pos	-
$\tilde{\mathbf{x}}$	the best and most unpredictable comedy this year	0.13	-	0.87
$\mathbf{x}$	we never feel anything for these characters	0.95	neg	-
$\tilde{\mathbf{x}}$	we never empathize for these characters	0.11	-	0.93

Fig. B.1. Hypothetical results of using *SEA* [96]. Note that unlike *HotFlip*, *SEA* can change and delete multiple tokens simultaneously as it samples from a paraphrasing model. Again, $\mathbf{x}$ indicates the original sentence, $\tilde{\mathbf{x}}$ indicates the adversarial sentence, and $S(\mathbf{x},\tilde{\mathbf{x}})$ is the semantical-equivalency-score which must be at least 0.8.

with a minimum semantical-equivalency-score and ensures the predicted label is different.

$$\begin{aligned} A_{\text{SEA}}(\mathbf{x}) = &\underset{\tilde{\mathbf{x}}\sim q(\tilde{\mathbf{x}}|\mathbf{x})}{\operatorname{argmax}} S(\mathbf{x},\tilde{\mathbf{x}}) \\ &\text{s.t. } S(\mathbf{x},\tilde{\mathbf{x}}) \geq 0.8 \\ &\quad \underset{i}{\operatorname{argmax}}\, p(i|\mathbf{x};\theta) \neq \underset{i}{\operatorname{argmax}}\, p(i|\tilde{\mathbf{x}};\theta) \\ &\text{where } S(\mathbf{x},\tilde{\mathbf{x}}) = \min\left(1, \frac{q(\tilde{\mathbf{x}}|\mathbf{x})}{q(\mathbf{x}|\mathbf{x})}\right). \end{aligned} \tag{22}$$

The reason why a relative score is necessary, as opposed to just using $S(\mathbf{x},\tilde{\mathbf{x}}) = q(\tilde{\mathbf{x}}|\mathbf{x})$, is that for two normal sentences $\mathbf{x}_1$ and $\mathbf{x}_2$ of different length, longer sentences are just inherently less likely. Therefore, to maintain a comparative semantical-equivalency-score normalizing by $q(\mathbf{x}|\mathbf{x})$ is necessary [96].

C COUNTERFACTUALS

C.1 Polyjuice

Polyjuice by Wu et al. [134] is primarily a *counterfactual dataset* generator, and the generation is therefore detached from the model. However, by strategically filtering these generated examples such that the model's prediction is changed the most, they condition the *counterfactual* generation on the model, thereby making a *post-hoc* explanation.

The generation is done by fine-tuning a GPT-2 model [90] on existing *counterfactual datasets* [41, 58, 74, 101, 130, 138]. For each pair of original and counterfactual example, they produce a training prompt, see Equation (23) for the exact structure. What the conditioning code is and what is replaced in Equation (23) is determined by the existing *counterfactual datasets*.

$$\begin{aligned} \textit{prompt} = \text{“}&\underbrace{\text{It is great for kids}}_{\text{original sentence}} \texttt{<GENERATE>} \\ &\underbrace{\texttt{[negation]}}_{\text{conditioning code}} \underbrace{\texttt{It is [BLANK] great for [BLANK]}}_{\text{masked counterfactual}} \\ &\texttt{<REPLACE>} \underbrace{\texttt{not [ANSWER] children [ANSWER]}}_{\text{masking answers}} \texttt{<EOS>”} \end{aligned} \tag{23}$$

For a *counterfactual* generation, they specify the original sentence and optionally the condition code, and then let the model generate the *counterfactuals*. These *counterfactuals* are independent of the model. To make them dependent on the model, they filter the *counterfactuals* and select those examples that change the prediction the most. One important detail is that they adjust the prediction change with an *importance measure* (*SHAP*), such that the *counterfactual examples* that could

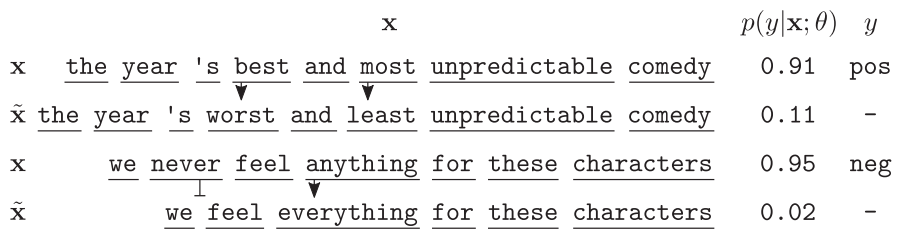

Fig. C.1. Hypothetical results of *Polyjuice*, showing how some words were either replaced or removed to produce *counterfactual examples*.

have been generated by an *importance measure* are valued less. An example of this explanation can be seen in Figure C.1.

To validate *Polyjuice*, for a *human-grounded* experiment, they show that humans were unable to predict the model's behavior for the *counterfactual examples*, thereby concluding that their method highlights potential robustness issues. Whether *Polyjuice* is *functionally-grounded* is somewhat questionable, because the model is not a part of the generation process itself, it is merely used as a filtering step.

D RULES

D.1 Compositional Explanations of Neurons

In *Compositional Explanations of Neurons* by Mu and Andreas [80], the rule generation problem is simplified by only relating the presence of input words to the activation of a single neuron.

The rules typically have the form of logical rules, meaning `not`, `and`, and `or`, where the Booleans indicate a word is present as seen in Figure D.1, although Mu and Andreas [80] do not make any hard constraints here. For example, in an NLI task they also have indicators for POS-presence and word-overlap between the hypothesis and premise. If these rules are satisfied it means the neuron activation is above a defined threshold. For example, in a ReLU(·) unit one can threshold if its post-activation is above 0.

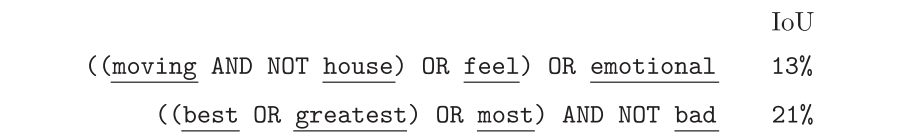

Fig. D.1. Hypothetical example showing rules which activates a selected neuron. IoU is how often the rule activated the neuron, compared to cases where either the rule is true or the neuron activated (higher is better).

Given a dataset $\mathcal{D}$, a neuron activation $z_n(\mathbf{x})$, a threshold τ, and an indicator function for the rule $R(\mathbf{x})$, the aggrement between the rule and the neuron activation can be measured with the *Intersection over Union score*:

$$\text{IoU}(n, R) = \frac{\sum_{x \in \mathcal{D}} \mathbb{1}(z_n(\mathbf{x}) > \tau \wedge R(\mathbf{x}))}{\sum_{x \in \mathcal{D}} \mathbb{1}(z_n(\mathbf{x}) > \tau \vee R(\mathbf{x}))}. \tag{24}$$

For one particular neuron n, the combinatorial rule R is then constructed using beam-search which stops at a pre-defined number of iterations. At each iteration, all feature indicator functions (e.g., word in $\mathbf{x}$) and their negative, combined with the logical operators `and` and `or`, are scored using $\text{IoU}(n, R)$.

Unfortunately, Mu and Andreas [80] do not perform any *groundedness* validation of this approach. Furthermore, as the method only looks at the relation between the input and the neuron, it is unclear how much the selected neuron affects the output.

REFERENCES

[1] Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 4190–4197. DOI: https://doi.org/10.18653/v1/2020.acl-main.385

[2] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. DOI: https://doi.org/10.1109/ACCESS.2018.2870052

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc., 9505–9515.

[4] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of the International Conference on Learning Representations*. 1–12.

[5] David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 412–421. DOI: https://doi.org/10.18653/v1/D17-1042

[6] Jacob Andreas, Anca Dragan, and Dan Klein. 2017. *Translating Neuralese*. Technical Report. 232–242 pages. DOI: https://doi.org/10.18653/v1/P17-1022

[7] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 39–48. DOI: https://doi.org/10.1109/CVPR.2016.12

[8] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus Robert Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11 (2010), 1803–1831.

[9] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*. 1–15.

[10] Naman Bansal, Chirag Agarwal, and Anh Nguyen. 2020. SAM: The sensitivity of attribution methods to hyperparameters. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 11–21. DOI: https://doi.org/10.1109/CVPRW50498.2020.00009

[11] Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2021. "Will you find these shortcuts?" A protocol for evaluating the faithfulness of input salience methods for text classification. arXiv:2111.07367. Retrieved from http://arxiv.org/abs/2111.07367.

[12] Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?. In *Proceedings of the 3rd BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Stroudsburg, PA, 149–155. DOI: https://doi.org/10.18653/v1/2020.blackboxnlp-1.14

[13] Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: a survey. *Transactions of the Association for Computational Linguistics* 7 (2019), 49–72. https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00254/43503/Analysis-Methods-in-Neural-Language-Processing-A.

[14] Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. 2020. Interpretability and analysis in neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Association for Computational Linguistics, Stroudsburg, PA, 1–5. DOI: https://doi.org/10.18653/v1/2020.acl-tutorials.1

[15] Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics* 7 (2019), 49–72. DOI: https://doi.org/10.1162/tacl_a_00254

[16] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, New York, NY, 610–623. DOI: https://doi.org/10.1145/3442188.3445922

[17] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. 2019. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 648–657. DOI: https://doi.org/10.1145/3351095.3375624

[18] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022. Retrieved from https://jmlr.org/papers/v3/blei03a.html.

[19] Tolga Bolukbasi, Kai Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems* (2016), 4356–4364.

[20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. Retrieved from https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[21] Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=BJg1f6EFDB.

[22] Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Michael Weigelt. 2019. Natural language multitasking analyzing and improving syntactic saliency of latent representations. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS'17)*, arXiv:1801.06024. Retrieved from http://arxiv.org/abs/1801.06024.

[23] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural language inference with natural language explanations. In *Proceedings of the Advances in Neural Information Processing Systems*. 9539–9549.

[24] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (7 2019), 832. DOI: https://doi.org/10.3390/electronics8080832

[25] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M. Rao, Troy D. Kelley, Dave Braines, Murat Sensoy, Christopher J. Willis, and Prudhvi Gurram. 2017. Interpretability of deep learning models: A survey of results. In *Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computed, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 1–6. DOI: https://doi.org/10.1109/UIC-ATC.2017.8397411

[26] Jonathan Chang, Jordan Boyd-graber, Sean Gerrish, Chong Wang, David M. Blei, Jordan Boyd-graber, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of the Advances in Neural Information Processing Systems*. Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Eds.), Vol. 22. Curran Associates, Inc., 288–296. Retrieved from https://proceedings.neurips.cc/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf.

[27] A. Chatzimparmpas, R. M. Martins, I. Jusufi, K. Kucher, F. Rossi, and A. Kerren. 2020. The state of the art in enhancing trust in machine learning models with the use of visualizations. *Computer Graphics Forum* 39, 3 (2020), 713–756. DOI: https://doi.org/10.1111/cgf.14034

[28] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. 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*. Association for Computational Linguistics, Stroudsburg, PA, 276–286. DOI: https://doi.org/10.18653/v1/W19-4828

[29] Louis Clouatre, Prasanna Parthasarathi, Amal Zouaq, and Sarath Chandar. 2022. Local structure matters most: Perturbation study in NLU. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022*, Association for Computational Linguistics, Stroudsburg, PA, 3712–3731. http://arxiv.org/abs/2107.13955 https://aclanthology.org/2022.findings-acl.293.

[30] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg, Ann Yuan Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg, Ann Yuan, Martin Wattenberg, Fernanda B. Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Proceedings of the Advances in Neural Information Processing Systems*. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 8594–8603. Retrieved from https://proceedings.neurips.cc/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf.

[31] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 2126–2136. DOI: https://doi.org/10.18653/v1/P18-1198

[32] R. Dennis Cook and Sanford Weisberg. 1980. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics* 22, 4 (1980), 495–508. DOI: https://doi.org/10.1080/00401706.1980.10486199

[33] Anna B. Costello and Jason W. Osborne. 2005. Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research and Evaluation* 10, 7 (2005), 1–9. DOI: https://doi.org/10.7275/jyj1-4868

[34] Charles B. Crawford and George A. Ferguson. 1970. A general rotation criterion and its use in orthogonal rotation. *Psychometrika* 35, 3 (1970), 321–332. DOI: https://doi.org/10.1007/BF02310792

[35] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, Suzhou, 447–459. https://aclanthology.org/2020.aacl-main.46.

[36] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 4171–4186.

[37] Finale Doshi-Velez and Been Kim. 2017. Towards A rigorous science of interpretable machine learning. arXiv:1702.08608. Retrieved from http://arxiv.org/abs/1702.08608.

[38] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Christopher Bavitz, Sam Gershman, David O'Brien, Kate Scott, Stuart Schieber, James Waldo, David Weinberger, Adrian Weller, and Alexandra Wood. 2017. Accountability of AI under the law: The role of explanation. *SSRN Electronic Journal* DOI: https://doi.org/10.2139/ssrn.3064761

[39] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Communications of the ACM* 63, 1 (2019), 68–77. DOI: https://doi.org/10.1145/3359786

[40] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 31–36. DOI: https://doi.org/10.18653/v1/P18-2006

[41] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Stroudsburg, PA, 1307–1323. DOI: https://doi.org/10.18653/v1/2020.findings-emnlp.117

[42] Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L. Alfonso Ureña-López. 2021. A survey on bias in deep NLP. *Applied Sciences* 11, 7 (2021), 3184. DOI: https://doi.org/10.3390/app11073184

[43] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, Vol. 32.

[44] Yash Goyal, Uri Shalit, and Been Kim. 2019. Explaining classifiers with causal concept effect (CaCE). arXiv:1907.07165. Retrieved from http://arxiv.org/abs/1907.07165.

[45] Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. 2021. FastIF: Scalable influence functions for efficient model interpretation and debugging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 10333–10350. https://aclanthology.org/2021.emnlp-main.808.

[46] Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *Proceedings of the International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=SygWvAVFPr.

[47] Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 5553–5563. DOI: https://doi.org/10.18653/v1/2020.acl-main.492

[48] Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language?. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Stroudsburg, PA, 4351–4367. DOI: https://doi.org/10.18653/v1/2020.findings-emnlp.390

[49] John Hewitt and Percy Liang. 2019. 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*. Association for Computational Linguistics, Stroudsburg, PA, 2733–2743. DOI: https://doi.org/10.18653/v1/D19-1275

[50] Daniel E. Ho and Alice Xiang. 2020. Affirmative algorithms: The legal grounds for fairness as awareness. *The University of Chicago Law Review Online*. http://arxiv.org/abs/2012.14285.

[51] Sara Hooker, Dumitru Erhan, Pieter-Jan Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*.

[52] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. 2019. Global explanations of neural networks. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, New York, NY, 279–287. DOI: https://doi.org/10.1145/3306618.3314230

[53] Alon Jacovi and Yoav Goldberg. 2020. 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*. Association for Computational Linguistics, Stroudsburg, PA, 4198–4205. DOI: https://doi.org/10.18653/v1/2020.acl-main.386

[54] Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, Stroudsburg, PA, 3543–3556. DOI: https://doi.org/10.18653/v1/N19-1357

[55] Myeongjun Jang and Thomas Lukasiewicz. 2021. Are training resources insufficient? Predict first then explain! arXiv:2110.02056. Retrieved from http://arxiv.org/abs/2110.02056.

[56] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv:2001.08361. Retrieved from http://arxiv.org/abs/2001.08361.

[57] Emilie Kaufmann and Shivaram Kalyanakrishnan. 2013. Information complexity in bandit subset selection. In *Proceedings of the Journal of Machine Learning Research*. Retrieved from http://proceedings.mlr.press/v30/Kaufmann13.pdf.

[58] Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *Proceedings of the International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=Sklgs0NFvr.

[59] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). *35th International Conference on Machine Learning*. 6 (2018), 4186–4195.

[60] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (Un)reliability of saliency methods. In *Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, Vol. 11700 LNCS, Springer, 267–280. http://link.springer.com/10.1007/978-3-030-28954-6_14.

[61] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*. 2976–2987.

[62] Arne Köhn. 2015. What's in an embedding? Analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 2067–2073. DOI: https://doi.org/10.18653/v1/D15-1246

[63] Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 8730–8742. DOI: https://doi.org/10.18653/v1/2020.acl-main.771

[64] Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation. arXiv:2004.05569. Retrieved from http://arxiv.org/abs/2004.05569

[65] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 107–117. DOI: https://doi.org/10.18653/v1/D16-1011

[66] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. 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*. Association for Computational Linguistics, Stroudsburg, PA, 681–691. DOI: https://doi.org/10.18653/v1/N16-1082

[67] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics* 4, 1990 (2016), 521–535. DOI: https://doi.org/10.1162/tacl_a_00115

[68] Zachary C. Lipton. 2018. The mythos of model interpretability. *Communications of the ACM* 61, 10 (2018), 36–43. DOI: https://doi.org/10.1145/3233231

[69] Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 5570–5581. DOI: https://doi.org/10.18653/v1/P19-1560

[70] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692. Retrieved from https://github.com/pytorch/fairseq.

[71] Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*. 4766–4775. http://arxiv.org/abs/1705.07874.

[72] Andreas Madsen. 2019. Visualizing memorization in RNNs. *Distill* 4, 3 (2019). DOI: https://doi.org/10.23915/distill.00016

[73] Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2021. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. arXiv:2110.08412. Retrieved from http://arxiv.org/abs/2110.08412.

[74] Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 3428–3448. DOI: https://doi.org/10.18653/v1/P19-1334

[75] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys* 54, 6 (2021), 1–35. DOI: https://doi.org/10.1145/3457607

[76] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems* 32 (2019), 1–13.

[77] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*. Retrieved from http://ronan.collobert.com/senna/.

[78] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38. DOI: https://doi.org/10.1016/j.artint.2018.07.007

[79] Christoph Molnar. 2019. *Interpretable Machine Learning*. Independent. 318 pages. Retrieved from https://christophm.github.io/interpretable-ml-book/.

[80] Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. In *Proceedings of the Advances in Neural Information Processing Systems*.

[81] Pramod K. Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question?. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 1896–1906. DOI: https://doi.org/10.18653/v1/p18-1176

[82] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453. DOI: https://doi.org/10.1126/science.aax2342

[83] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Morristown, NJ, 311. DOI: https://doi.org/10.3115/1073083.1073135

[84] Sungjoon Park, JinYeong Bak, and Alice Oh. 2017. Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 401–411. DOI: https://doi.org/10.18653/v1/D17-1041

[85] Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 411–420. DOI: https://doi.org/10.5555/2074022.2074073

[86] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. DOI: https://doi.org/10.1080/14786440109462720

[87] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 1532–1543. DOI: https://doi.org/10.3115/v1/D14-1162

[88] Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. 2020. Estimating training data influence by tracing gradient descent. In *Proceedings of the Advances in Neural Information Processing Systems*.

[89] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI* (2018). Retrieved from https://openai.com/blog/language-unsupervised/.

[90] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9. Retrieved from https://openai.com/blog/better-language-models/.

[91] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21 (2020), 1–67. Retrieved from https://jmlr.org/papers/v21/20-074.html.

[92] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! Leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 4932–4942. DOI: https://doi.org/10.18653/v1/P19-1487

[93] Karthikeyan Natesan Ramamurthy, Bhanukiran Vinzamuri, Yunfeng Zhang, and Amit Dhurandhar. 2020. Model agnostic multilevel explanations. arXiv:2003.06005. Retrieved from http://arxiv.org/abs/2003.06005.

[94] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1135–1144. DOI: https://doi.org/10.1145/2939672.2939778

[95] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 1527–1535. Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982.

[96] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 856–865. DOI: https://doi.org/10.18653/v1/P18-1079

[97] Marko Robnik-Šikonja and Marko Bohanec. 2018. *Perturbation-Based Explanations of Prediction Models*. Springer International Publishing. 159–175 pages. DOI: https://doi.org/10.1007/978-3-319-90403-0_9

[98] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics* 8 (2020), 842–866. DOI: https://doi.org/10.1162/tacl_a_00349

[99] Alexis Ross, Ana Marasović, and Matthew E. Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, Stroudsburg, PA, 3840–3852. https://aclanthology.org/2021.findings-acl.336.

[100] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215. DOI: https://doi.org/10.1038/s42256-019-0048-x

[101] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial Winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (2020), 8732–8740. DOI: https://doi.org/10.1609/aaai.v34i05.6399

[102] Amit Sangroya, Mouli Rastogi, C. Anantaram, and Lovekesh Vig. 2020. Guided-LIME: Structured sampling based hybrid approach towards explaining blackbox machine learning models. In *Proceedings of the CEUR Workshop*.

[103] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. 2001. A generalized representer theorem. In *Proceedings of the International Conference on Computational Learning Theory*. Springer, 416–426. DOI: https://doi.org/10.1007/3-540-44581-1_27

[104] Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 2931–2951. DOI: https://doi.org/10.18653/v1/P19-1282

[105] Shapley. 1953. A value for N-Person games. *Contributions to the Theory of Games (AM-28), Volume II* (1953), 307–317. Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/604084.pdf.

[106] Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax?. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 1526–1534. DOI: https://doi.org/10.18653/v1/d16-1159

[107] Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. UnNatural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. http://arxiv.org/abs/2101.00010.

[108] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM, New York, NY, 180–186. DOI: https://doi.org/10.1145/3375627.3375830

[109] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 455–465. Retrieved from https://aclanthology.org/P13-1045/.

[110] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1631–1642.

[111] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. 2019. Seq2seq-Vis: A visual debugging tool for sequence-to-sequence models. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 353–363. DOI: https://doi.org/10.1109/TVCG.2018.2865044

[112] Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 5594–5608. DOI: https://doi.org/10.18653/v1/2020.acl-main.495

[113] Xiaofei Sun, Diyi Yang, Xiaoya Li, Tianwei Zhang, Yuxian Meng, Han Qiu, Guoyin Wang, Eduard Hovy, and Jiwei Li. 2021. Interpreting deep learning models in natural language processing: A review. arXiv:2110.10470. Retrieved from http://arxiv.org/abs/2110.10470.

[114] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*. 5109–5118.

[115] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, Stroudsburg, PA, 4149–4158. DOI: https://doi.org/10.18653/v1/N19-1421

[116] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 4593–4601. DOI: https://doi.org/10.18653/v1/P19-1452

[117] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Stroudsburg, PA, 107–118. https://www.aclweb.org/anthology/2020.emnlp-demos.15.

[118] 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. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *Proceedings of the 7th International Conference on Learning Representations*. 1–17. Retrieved from https://openreview.net/forum?id=SJzSgnRcKX.

[119] Erico Tjoa and Cuntai Guan. 2020. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems* 14, 8 (2020), 1–21. DOI: https://doi.org/10.1109/TNNLS.2020.3027314

[120] Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. Retrieved from https://www.jmlr.org/papers/v9/vandermaaten08a.html.

[121] Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across NLP tasks. arXiv:1909.11218. Retrieved from http://arxiv.org/abs/1909.11218.

[122] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Proceedings of the Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 12388–12401. Retrieved from https://proceedings.neurips.cc/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

[123] Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 183–196. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.14

[124] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, 2153–2162. DOI: https://doi.org/10.18653/v1/D19-1221

[125] Eric Wallace, Matt Gardner, and Sameer Singh. 2020. Interpreting predictions of NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*. Association for Computational Linguistics, Stroudsburg, PA, 20–23. DOI: https://doi.org/10.18653/v1/2020.emnlp-tutorials.3

[126] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=rJ4km2R5t7.

[127] Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. 2021. Towards a robust deep neural network against adversarial texts: A survey. *IEEE Transactions on Knowledge and Data Engineering* 1–1. https://ieeexplore.ieee.org/document/9557814/.

[128] Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21)*. http://arxiv.org/abs/2102.12060.

[129] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 11–20. DOI: https://doi.org/10.18653/v1/D19-1002

[130] John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 451–462. DOI: https://doi.org/10.18653/v1/P18-1042

[131] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Stroudsburg, PA, 1112–1122. DOI: https://doi.org/10.18653/v1/N18-1101

[132] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z. Gajos, Walter S. Lasecki, and Neil Heffernan. 2016. AXIS. In *Proceedings of the 3rd (2016) ACM Conference on Learning @ Scale*. ACM, New York, NY, 379–388. DOI: https://doi.org/10.1145/2876034.2876042

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

[134] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021. 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)*, Association for Computational Linguistics, Stroudsburg, PA, 6707–6723. https://aclanthology.org/2021.acl-long.523.

[135] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I. Inouye, Pradeep K. Ravikumar, Arun Sai Suggala, David I. Inouye, and Pradeep K. Ravikumar. 2019. On the (In)fidelity and sensitivity of explanations. In *Proceedings of the Advances in Neural Information Processing Systems 32*. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Curran Associates, Inc., Vancouver, Canada, 10967–10978. Retrieved from http://papers.nips.cc/paper/9278-on-the-infidelity-and-sensitivity-of-explanations.pdf.

[136] Chih-Kuan Yeh, Joon Sik Kim, Ian E. H. Yen, and Pradeep Ravikumar. 2018. Representer point selection for explaining deep neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*. 9291–9301.

[137] Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Stroudsburg, PA, 359–361. DOI: https://doi.org/10.18653/v1/W18-5448

[138] Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, Stroudsburg, PA, 1298–1308. DOI: https://doi.org/10.18653/v1/N19-1131

Received 14 October 2021; revised 18 June 2022; accepted 20 June 2022