# Interpretation of Black Box NLP Models: A Survey

**Shivani Choudhary, Niladri Chatterjee, Subir Kumar Saha**
{shivani@sire, niladri@maths, saha@mech}.iitd.ac.in
Indian Institute of Technology Delhi
Hauz Khas, Delhi-110016, India

## Abstract

Machine learning (ML) models are ubiquitous in every wake of life and employed in high-stakes decision-making tasks. The complexity involved in ML models, specifically in Neural models, has brought in the question, *"How does a model make a decision?"*. An insight into the model's decision-making process will help fix the accountability for the model's decision. It will further address ethical, safety, and bias issues associated with the model. To address these interpretability can provide an explanation that humans can easily understand. This survey will present a comprehensive survey of methods employed for interpretability. Firstly, it presents the definition of *Interpretability* followed by a discussion on the different approaches adopted in NLP space. Finally, It will highlight the synergy between different approaches and issues with interpretability in NLP space.

## 1 Introduction

Machine learning (ML) is now ubiquitous in the current era. Some of the ML models, especially deep learning-based models, achieve near-human accuracy. It has led to the adoption of ML into several areas, including critical areas like health, financial markets, criminal justice. (Lipton, 2016). It had raised an important question regarding the black-box nature of the ML models.

There is no concrete definition of the Interpretability of ML. However, different authors have tried to provide some definitions. Interpretability is often considered along with explainability; even some of the authors have used it interchangeably (Zhang et al., 2020). In a general sense, interpreting data means extracting information from them (Murdoch et al., 2019). It can be defined as *"the degree of understanding of a model in regards to how a decision was made so that a model can provide an answer to a user"*. It can also be understood as the extraction of the relevant knowledge from the ML models that was *"learned by the model"* or was *"present in the data"* (Murdoch et al., 2019). It highlights the key bit of interpretability, a user should understand and reason the model output. Learned understanding can be presented in different forms like visualization, decision tree, natural language and graph as well. Interpretability is not limited to model parameters, learning algorithm, feature selection, or a combination of these (Chakraborty et al., 2017; Doshi-Velez and Kim, 2017a). (Lipton, 2016) has listed *trust, causality, transferability and informativeness* as key elements when considering the interpretability research.

Most of the works in this area were focused on model working, on a global level or on a local level. However, only a handful of work tried to analyze the aspect of interpretability from a "user" prospect. In other words, they were answering the question "Interpretable to Whom?" because different levels of audience will interpret the model separately. E.g., A person with a Statistics major can interpret the behavior of Bayesian models to some extent, while a person who is not a domain expert may not find the exact information relevant for interpretation. So, It gives another dimension to interpretability-related research.

Machine learning (ML) models are sensitive to the perturbation in the input. Sometimes, even a minor change can lead to a change in prediction, let alone the confidence of the prediction. This behavior of ML models can also help to get an explanation for a prediction. In the case of computer vision, it can help understand which pixels or superpixels can lead to a behavior change.

This survey aims to present the theoretical overview of interpretability along with different interpretability methods. In section – 2, we present an overview of the need for interpretability. Section – 3 lists the classification of interpretability. Section – 4 lists different methods of interpretability with their application in the NLP task. This section also lists some of the debate surrounding those models. Finally, Section – 5 presents some of the findings with respect to different models followed by Section – 6 as conclusion. Table – 1 presents the categorization of models with a list of representative papers with the area of application in NLP.

# 2 Background

## 2.1 Why do we need interpretable models?

With broader adoption of the ML models in the various day-to-day interaction, it has brought in the aspect where a human needs to understand the behavior of ML models—it necessities a person to understand the process by which it has reached a particular conclusion. Engagement of ML models in activities like criminal recidivism, loan approval, premium calculation, etc., has brought in the ethical and fairness concerns in the ML adoption in real life. In order to understand or allay the apprehensions raised on the ML, there needs a requirement to understand the decision of the models.

### 2.1.1 Reliability

Adoption of models in different areas does not require it to be reliable every time. But, the scenarios where a decision outcome from the models can have a big impact need to be reliable. In case of medical diagnosis, like detection of *malignant* and *benign* tumour. By just changing a small set of pixels can lead to an altered output from a trained DL network (Finlayson et al., 2018). Another adversarial example (Wu et al., 2018) that leads to an incorrect prediction from the network from red light to green light. These predictions can cost a human life. So in these cases, reliability is required. Had the models been interpretable, the model's decision could have been explained. Specially, in case of an adversarial attacks (Ebrahimi et al., 2017).

In another scenario, Husky vs. Wolf classifier, the models seem to have learned the snow patch to make the classification (Ribeiro et al., 2016a). It has nothing to do with the features of the two breeds. In this case, due to some bit to interpretability of the decision outcome, we can understand the decision-making process. Though the predicted classes are incorrect, we know that this classifier has learned some irrelevant patterns and is not reliable.

### 2.1.2 Ethical concerns

In recent times, it has been brought to focus that ML models are biased due to bias ingrained in the dataset or due to algorithmic complexity. ProPublica has presented an analysis where it has shown that COMPASS, which predicts criminal recidivism, has a bias towards native African American. In another case, Amazon's one-day delivery was unavailable to the minority neighborhood while it was available in other neighborhoods. All these concerns stress on the fact that we need interpretable models through which we can understand the decision outcomes. Whether the decision was made using protected attributes like gender, place of birth, caste, etc.

### 2.1.3 Research aspects

Ml models learn different patterns from the data. The learned rules are a way to understand the unknown. Interpretability of the models will help us understand those undefined rules to understand the missing elements in fields like Physics, Genomics, etc.

### 2.1.4 Transparency

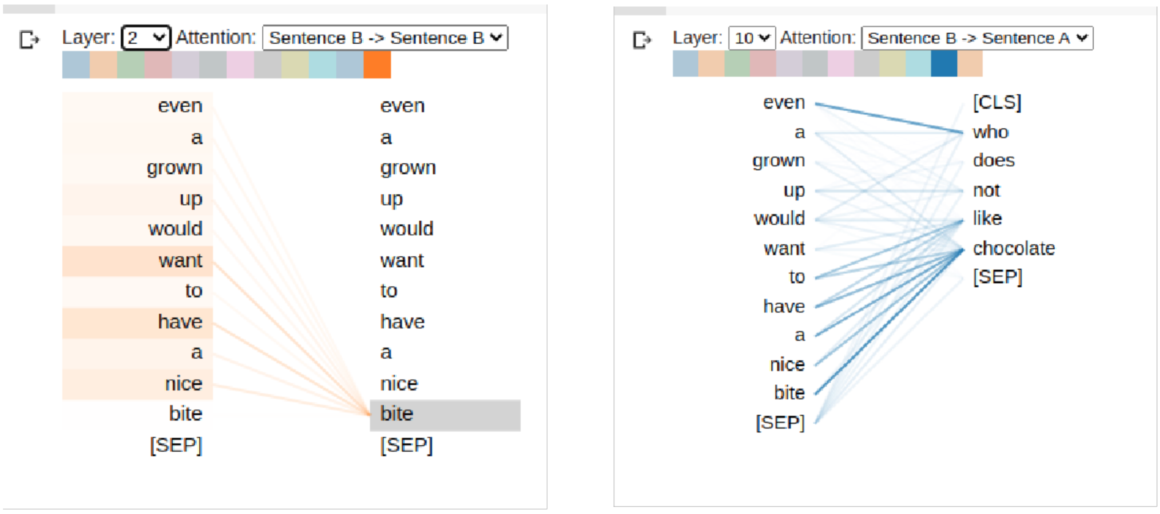
ML models are treated as a "black-box model". Apart from those parameters and associated weights increase the model's complexity. A decision outcome generated from those models is hard to comprehend. With an interpretable model, we can easily understand the working and evaluate whether the model characteristics like causality, transferability, and informativeness (Lipton, 2016). It will ensure that we would be able to evaluate the working of models in the case of real-world data. Then we can answer the questions from causality prospects like "what if?", "why?". It will enable a user to evaluate the model from counterfactual criteria.

# 3 Interpretability classification

Interpretability of a model can be addressed by several approaches depending upon the type of end explanation it generates. Explanation methods can be categorized based upon different scopes and properties.

## 3.1 Local vs. Global interpretability

Interpretability in this context can be understood as what part of the model can be interpreted. How much insight the generated explanation can provide.



Layer: 2 Attention: Sentence B -> Sentence B

Layer: 10 Attention: Sentence B -> Sentence A

(a) Attention from Sentence B to B, generated using (Vig, 2019) (b) Attention from Sentence B to A, generated using (Vig, 2019)

#### 3.1.1 Global Interpretability

A model can be categorized as globally interpretable if we can comprehend a model's behavior altogether from the parameter, weights, and other ancillary information (Lipton, 2016). It demands the understanding to the extent to which a user can figure out the interplay between features and the weightage of features. Global interpretability centers on how well parametric variation and association can be comprehended by humans. In real-life models, there are a huge number of parameters associated with the algorithms. The complexity of the global interpretation is added by the number of features present in the data. With a lot of features in hand, it is difficult to interpret a model globally (Honegger, 2018; Cowan, 2010). In such cases, we need to employ algorithms like t-SNE to present higher dimension data points in a lower-dimensional space (van der Maaten and Hinton, 2008).

In this class of models are decision tree, linear regression, rule extractor, where we can either interpret the rule from plain text, by the split at the nodes or by examining the weights.

#### 3.1.2 Local Interpretability

Local interpretability is more focused on understanding the system's behavior concerning a single data sample or a group of samples in the neighborhood. A complex model is usually comprehended by an approximate linear model. A good approximation can be achieved with a sacrifice of accuracy for this model. In this case, the model's behavior can be analyzed for a single case or a group case. When a group of data points is considered then, approximate will be performed the local interpretation task for each of the samples (Molnar, 2019).

### 3.2 Inherently interpretable vs Post-hoc explanation

Models can be broadly categorized in the Inherently interpretable or Post-hoc explanation-based interpretable model (Doshi-Velez and Kim, 2017a). Inherently interpretable models are those which are by design interpretable. Due to constraints, these models are often mentioned as "white-box models" (Rudin, 2018). These models are easy to interpret. As (Jacovi and Goldberg, 2020) has pointed out that a model may not be interpretable just by claiming it to be interpretable. It needs to be verified, as is the case with *Attention mechanism* in NLP.

Interpretation via post-hoc techniques is applied after the model is trained. The weights are interpreted using different local and global analysis techniques. Post-hoc explanation can be applied to the inherently interpretable models as well.

In the above two interpretations, we are concerned with the model's behavior. Another aspect of interpretability can be driven from the data itself, even before it is consumed into the ML models/algorithm. Some definitive trends can be extracted from the data by doing exploratory analysis. This pre-training information helps to interpret the models.

### 3.3 Model specific and model agnostic

Post-hoc explanations can be categorized into two parts based on their application and relevance to the models' internals. Model-specific methods are dependent on the internals of the model. These types of methods can be applied to a specific family of models. On the other hand, model agnostic methods can be applied to any model. Model agnostic methods consider any ML models as black box. So it does not have any information about the internal organization of the model (Molnar, 2019).

### 3.4 NLP Specific

Interpretability in natural language processing (NLP) is an important aspect of understanding why a response to a specific question was made. We can take sentiment analysis as an example. Under this, a model classifies a sentence as positive or negative sentiment based upon the sentence. In order to explain the outcome, we need to understand different features of a sentence like tokens (word may be in original form or base form), syntactical structure, punctuation, etc.

The analysis would require finding the token that has carried more weightage than others in the decision-making process. BERT (Devlin et al., 2019) is based on *Attention mechanism* which is considered as inherently interpretable in nature. A visualization of attention between two sentences from different layers gives an insight into how each word influences each word. As an example, two sentences are taken as input *"Who does not like chocolate"* and *"Even a grown-up would want to have a nice bite"*. Figure-1a and Figure-1b shows the different level of influence between the words. Interpretability of the models would provide different types of explanation. Visualization, a summary of feature importance are one of those examples.

In order to address the interpretability, some aspects that can be probed will be explained in the survey are listed below. These are further categorized into two parts Local and Global explanation.

- Local Explanation

  **Feature based:** These methods focus on the task; what are the essential features that have impacted the model's decision outcome? It is further divided into different methods based on the type of model's feature/data point it uses to generate an explanation.

  **Causality-based scenario:** How would model scenario when it is tested from the Adversarial and counterfactual viewpoint? It evaluates the model robustness and performance in an alternate scenario.

  **Natural Language Explanations:** This approach generates explanation that can be understood by a layman.

- Global Explanation

  **Visualization:** How does each word influence each other in the model's definition?

  **Probing:** It tries to analyse and evaluate, “what are the linguistic features that are captured by the model?"

## 4 Interpretability Methods

### 4.1 Feature based

#### 4.1.1 Gradient based methods

Simonyan et al. (Simonyan et al., 2014) has proposed using the gradient of the output with respect to pixels of an input image to compute a “saliency map" of the image in the context of image classification tasks. In NLP domain, it is transformed into taking the gradient of output logits with respect to input. It measures the effect of change in input to the output generated by the model. Common methods based on the use of gradients are DeepLift (Shrikumar et al., 2017), Layerwise relevance propogation (LRP) (Binder et al., 2016), Guided-back propagation (Springenberg et al., 2015), deconvolutional networks (Zeiler et al., 2010). Gradient extraction helps to identify the important features for a given prediction. These method faces issue with sensitivity and implementation invariance. It means if two inputs with one differentiating feature (token) leads to a change in prediction then it needs to treated as an important feature. Sundararajan et.al. (Sundararajan et al., 2017) proposed Integrated gradient (IG) method which can address the issue of sensitivity and implementation invariance. IG works on the approach where gradients are accumulated for all the points on a straight line between an input and a baseline point. He et. al. (He et al., 2019) has applied this approach in neural machine translation (NMT) task and Mudrakarta (Mudrakarta et al., 2018) applied it in Question-Answering task to understand the keywords which were influencing the answers. Arras et.al. (Arras et al., 2016) has used LRP to analyse CNN trained for topic categorization task. Wang et. al. (Wang et al., 2020) has shown that gradient based analysis can be manipulated.

| Methods | Approach | Type of Analysis | Representative Paper | NLP domain Application |
|---|---|---|---|---|
| Local Interpretability | | | | |
| Feature based | Gradient based | Model agnostic | (Shrikumar et al., 2017)<br>(Binder et al., 2016)<br>(Springenberg et al., 2015)<br>(Zeiler et al., 2010) | NMT, QA,<br>Topic classification |
| | Input perturbation | Model agnostic | (Ribeiro et al., 2016b)<br>(Ribeiro et al., 2018a)<br>(Rychalska et al., 2018) | QA |
| | SHAP | Model agnostic | (Lundberg and Lee, 2017)<br>(Lundberg et al., 2018) | QA,<br>Text classification |
| | Attention based | Inherently interpretable and model specific[1] | (Tu et al., 2020)<br>(Lu et al., 2016)<br>(Li et al., 2020)<br>(Mao et al., 2019) | QA, VQA<br>Sentiment Analysis |
| Causality based | Adversarial examples | Model Specific | (Ebrahimi et al., 2018)<br>(Ribeiro et al., 2018b)<br>(Sato et al., 2018) | NMT, VQA,<br>Sentiment Analysis,<br>Grammatical error detection |
| | Counterfactual explanation | Model agnostic<br>and<br>Model specific | (Wu et al., 2021a)<br>(Ross et al., 2021a)<br>(Raffel et al., 2020)<br>(Elazar et al., 2021a)<br>(Vig et al., 2020)<br>(Finlayson et al., 2021) | Bias in model,<br>Syntactic evaluation<br>POS |
| NLE | Natural language explanation | Model Specific | (Park et al.)<br>(Ling et al., 2017)<br>(Tim et al., 2018)<br>(Kumar and Talukdar, 2020)<br>(Mccann et al., 2019) | NMT,<br>Label prediction,<br>Natural language inference |
| Global Interpretability | | | | |
| Visualization | Visualization | Model Agnostic | (Park et al., 2017)<br>(Li et al., 2016)<br>(Shin et al., 2018) | Linguistic features |
| Probing | Distributional word embedding probing | | (Ebrahimi et al., 2018)<br>(Ribeiro et al., 2018b)<br>(Sato et al., 2018) | NMT, VQA,<br>Sentiment Analysis,<br>Grammatical error detection |
| | Hidden state probing | Model agnostic<br>and<br>Model specific | (Shi et al., 2016)<br>(Belinkov et al., 2017)<br>(Mareek et al., 2020)<br>(Conneau et al., 2018)<br>(Hupkes and Zuidema, 2018)<br>(Peters et al., 2020) | NMT,<br>Compositionality,<br>Correference resolution<br>syntactic feature |

Table 1: List of interpretability methods in NLP. This table is separated in two parts – global method or local method.

*Note-1:* Method column presents the broader classification of methods. Second column presents the fine categorization under the broader category. Last column lists down NLP task employed to interpret the model's behaviour. Type of Analysis is an approximate categorization by taking the features used by Interpretability method.

*Note-2: It is not an one-to-one mapping with the representative paper.Details of each broader classification is presented in section-4. List of papers is not exhaustive. There are some of the papers that are related to different methods are listed in the discussion.*

### 4.1.2 Input Perturbation Based

It is another method to extract the importance of the different features present in the input samples. In this method, a word (token) or a collection of words (tokens) are modified or removed from the input samples, and a resulting change is measured. Feature importance is measured by the drop in the performance of the model. If the drop is high then the feature is very important for the model. These methods are model agnostic in nature.

Ribeiro et.al. (Ribeiro et al., 2016b) proposed a locally interpretable and model agnostic explanation (LIME) framework. In this method the model under consideration is assumed as a black box model. The central idea of LIME is to generate a local surrogate, a glass-box model, to generate explanations for the decision outcomes. LIME generates a dataset with perturbed inputs and corresponding predictions from the black box model. On this new dataset, LIME trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest. Mathematical formulation of LIME as below

$$\xi(x) = \underset{g\in\mathcal{G}}{\operatorname{argmax}}\, \mathcal{L}(f,g,\pi_x) + \Omega(g) \qquad (1)$$

- f(x) is the prediction from the black model for sample x
- $\mathcal{G}$ is the class of potential interpretable models
- $\pi_x$ defines the size of the neighbourhood
- $\mathcal{L}$ is the Loss that will measure the closeness of explanation
- $\Omega(g)$ determines the complexity of the local surrogate models

In original LIME method the analysis is based on the word level (single token), later they proposed a new model that is based on consecutive tokens (Ribeiro et al., 2018a). Consecutive tokens are called as *Anchors*. Anchors explains individual predictions of any black-box classification model by finding a decision rule that "anchors" the prediction sufficiently. A rule anchors a prediction if changes in other text does not change the prediction. LIME out can vary significantly even if two artificial points are in proximity (Alvarez-Melis and Jaakkola, 2018) and Slack et. al. pointed out that it is prone to adversarial attacks (Slack et al., 2020; Tan et al., 2019). Different version of LIME are proposed Zafar et. al. proposed D-LIME (Zafar and Khan, 2019), Zhou et.al. proposed S-LIME (Zhou et al., 2021).

LIME has been employed in QA task by Basaj et.al. (Rychalska et al., 2018) to check how many words from question are relevant to predict correct answer. Sydorova et.al. has applied it for QA task in conjugation with knowledge base(Sydorova et al., 2019a).

### 4.1.3 SHAP

SHAP (*SH*apley *A*dditive ex*P*lanations) was proposed by Lundberg and Lee (Lundberg and Lee, 2017). It is based on game theory based Shapely Values (Shapley, 1953). Methods like LIME may not distribute attributions fairly among the features while Shapely value guarantees it (Molnar, 2019). A way to use the efficient distribution using Shapely value would be to compute shapely values for each and every combination of the features ( a power set of the features) by training a linear model. But, it will be computationally expensive to train $2^M$ models for M set of features.

SHAP calculates Shapely value and presents it as a linear model or additive feature attribution. SHAP presents a model explanation as

$$g(x') = \phi_0 + \sum_{j=1}^{\mathcal{M}} \phi_j z'_j \qquad (2)$$

Where $g$ is an explanation model, $\mathcal{M}$ is the maximum size of coalition, $\phi_j$ is the feature attribution for feature $j$ and $z'$ is the binary vector.

The model agnostic version of the SHAP is Kernel SHAP. Lundberg and Lee call it as LIME + Shapely values. The solution of equation 1 will satisfy the property of local accuracy, missingness and consistency (Lundberg and Lee, 2017). Finding those values heuristically would be problematic. They suggested following would be choice for the parameters in equation 1.

$$\Omega(g) = 0$$

$$\pi_x(z') = \frac{(M-1)}{(Mchoose|z'|)|z'|(M-|z'|)}$$

$$\mathcal{L}(f,g,\pi'_x) = \sum_{(z'\in Z)} [f(h_x(z') - g(z')]^2 \pi_x(z')$$

where $|z'|$ is the number of non-zero elements in $z'$, $h_x(z)$ is the mapping function that maps the combination $z'$ to original feature space. DeepSHAP (DeepLIFT + Shapely values) is a model

[1] A detail discussion whether Attention is inherently interpretable or not

specific version of SHAP. Lundberg et. al. (Lundberg et al., 2018) proposed TreeSHAP for tree based machine learning models. TreeSHAP has issues with giving importance to the non-important features as well. Some of the authors have also highlighted that SHAP is prone to adversarial attack (Slack et al., 2020).

In NLP, Zhao et. al. (Zhao et al., 2020) has developed SHAP to explain CNN based text classification model. Balouchzahi et. al. (Balouchzahi et al., 2021) has used it for fake news profiling using SHAP based feature selection. Wu et. al.(Wu et al., 2021b) has used it to generate counterfactual.

### 4.1.4 Attention based

Attention mechanism was proposed by Bahdanau et. al. (Bahdanau et al., 2015). Attention is a weighted sum of the intermediate representation in neural network. Attention weights from the attention based models can be used for local interpretation. Attention mechanism has gained traction in NLP task. It is a state-of-art architecture for the NLP task likes question answering, Neural machine translation, Visual question answering etc. In NLP context, the feature (token) with higher weight is considered as an important feature. Attention has been applied to question answering task (Tu et al., 2020; Sydorova et al., 2019b; Shen et al., 2018), dialogue suggestion system (Li et al., 2020) and sentiment analysis (Mao et al., 2019; Yan et al., 2021; Luo et al., 2018). This method has been applied to multimodal data (Lu et al., 2016) like visual question answering. It uses both text and image mode of data as input. Lu et. al. (Lu et al., 2016) called it *co-attention* where the reasoning is performed with question attention and visual attention. Combining the attention weights with visualization helps to interpret the model.

Several authors has used *Attention* in different NLP task. But there is an ongoing debate "*Is attention interpretable*" (Pruthi et al., 2020; Serrano and Smith, 2019; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Vashishth et al., 2019). (Jain and Wallace, 2019) and (Vashishth et al., 2019) has presented two arguments.

1. Attention weight should correlate with feature importance similar to gradient based methods
2. Alternative attention weights (counterfactual) should lead to changes in the prediction

Both the premise were not fulfilled in their experiments on question answering task and Natural language inference task. on the other hand (Serrano and Smith, 2019) had found that alternative weights did not necessarily resulted in outcome change. However, these arguments were countered by (Wiegreffe and Pinter, 2019) and argued that model's weight are learned in a unified manner with their parameters. So, detaching attention score from parts of the model will degrade the model itself. They also argued that *Attention* is not the only explanation. (Vashishth et al., 2019) has performed experiments on tasks like text classification, Natural language inference (NLI) and NMT, and concluded that the model's performance is dependent on the type of task. Attention weights are interpretable and correlate with feature importance — when weights are computed using two sequence which are the function of input and Attention weights may not be interpretable when the score is calculated on single sequence like Text classification.

## 4.2 Visualization

Visualization is an important way to understand how a neural model work (Li et al., 2016). It can be applied with any of the feature importance based methods. With visualization, we can project the feature importance weights using heatmap, partial dependency plot etc. Most of the state-of-art NLP task are dependent on the word embedding. Sparse encoding like one-hot encoding has been replace by dense encoding like (word2vec(Mikolov et al., 2013), Glove (Pennington et al., 2014) and representation from intermediate layers of BERT (Devlin et al., 2019) and ElMO (Peters et al., 2018)). Word embedding based information captures information at model level. Hence it presents the information at global level. It presents which type of linguistic features are learnt by the model. Dense word embedding is presented in hyper-space. In order to understand the embedding, it needs to projected into two or three dimensional space.
t-SNE (van der Maaten and Hinton, 2008) and principal component analysis are two important tools to present a high dimensional representation to a lower dimensional space. (Li et al., 2016) has presented how individual components get activated by adding negative and positive words to a sentence. Using t-SNE, they have also shown that neural model learn the properties of local composi-

[2] https://radimrehurek.com/gensim/auto_examples/tutorials/run_fasttext.html

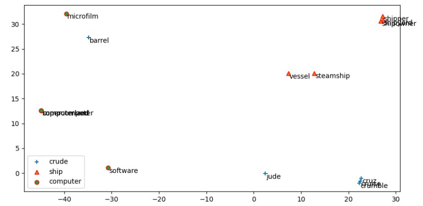


Figure 2: t-SNE 2D projection of FastText embedding [2]trained for 50 epoch on Reuters news corpus from NLTK, with context len 15

tionality, clustering negation+positive words (‘not nice’, ‘not good’) together with negative words. (Park et al., 2017) has suggested that rotation of the embedding can lead to an increased intepretability. His method is based on exploratory factor analysis and used PCA to visualize the representation. (Shin et al., 2018) has presented Eigen vector based method to analyse word embedding. An example of t-sne projection of FastText (Bojanowski et al., 2017) embedding of 5 nearest words to the words crude, ship and supercomputers in Figure-2.

## 4.3 Probing

With dependence of state-of-art NLP models on dense vector representation of the words (Devlin et al., 2019; Mikolov et al., 2013; Peters et al., 2018), it is pertinent to ask question like which type of linguistic features are encoded in the Word embedding or in the intermediate representation of the Neural models (Rogers et al., 2020; Zhang and Bowman, 2019; Poliak et al., 2020). A method to extract these information is called as probing. Probing tasks are commonly known as “auxillary tasks" (Adi et al., 2017), “diagnostic classifier" (Giulianelli et al., 2018; Hupkes and Zuidema, 2018) or decoding. Under this task, an external classifier is trained on the intermediate representation or word embedding to predict the liguistic property under the observation. It provides an insight into the fact that how well a model has learned the specified linguistic property. Several linguistic features have been analysed to extract different properties like Morphological, syntactic and semantic(Voita and Titov, 2020; Tang et al., 2021). It is based on the premise that if there is more task relevant information is learned by the model then it is going to perform better on the presented task. However, some researchers have pointed out that probe selection and measurement should be carefully done in order to get a reliable insight into the model (Voita and Titov, 2020; Hewitt and Manning, 2019). Linguistic insight can be extracted from the representations from intermediate layer of a neural model and word embedding. In this survey, we divide the probing into two parts – Hidden state probes and Distributional word Embedding probes.

### 4.3.1 Distributional word Embedding probes

Early word embedding methods were based on the distributional hypothesis, meaning information captured by a word can be extracted by the negihbourhood in which it was present. Two notable distributional word embedding CBOW/Skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Several author has tried to extract the information contained in the word embedding using simple classifiers like – logistic or linear classifiers. They all have reported the presence of the linguistic concepts to a varied extent (Köhn, 2016; Utsumi, 2020; Rubinstein et al., 2015; Gupta et al., 2015). (Ghannay et al., 2016) has further extended the probing using neural network for POS tagging, Named entity recognition and Mention detection. Apart from word embedding, sentence embedding based analysis has also been attempted by (Adi et al., 2017) using LSTM. They have noted the effectiveness of CBOW in LSTM based encoder task. (Conneau et al., 2018) and (Tenney et al., 2019) has presented a exhaustive list of task for the evaluation. This list was further extended by (Sorodoc et al., 2020; Şahin et al., 2020).

### 4.3.2 Hidden state probing

Distributional embedding has been replaced by contextual embedding like BERT (Devlin et al., 2019), ElMO(Peters et al., 2018) for NLP task. These embedding represent words as representations learned from the hidden states. Before we presents the work that examined deep-contextual embedding, we would like to highlight the works that has analysed the hidden state learned by neural models for different task. (Shi et al., 2016) has used probing technique in NMT task to determine whether LSTM based encode-decoder architecture can learn the syntactic features. They employed logistic classifier (as a diagnostic classifier) to predict different syntactic labels on top a learned sentence encoding vector and word by word hidden vectors. They pointed out that LSTM based encoder-decoder was able to learn different syntactic feature from the input sentence and different layers learned different features. They probed the model for 5 features

3 sentence level features — Voice, Tense and Top level syntactic sequence and 2 word level features — Parts of speech and smallest phrase constituent. (Belinkov et al., 2017) has extended the work of Shi et. al. for NMT task.(Raganato, 2018) has used probing for attention based models and (Mareek et al., 2020) has extended for multi-lingual task based on the (Conneau et al., 2018) 10 linguistic task for probing.

Use of probing is not limited to NMT task only. (Hupkes and Zuidema, 2018) has used it to find the learning capabilities of neural model in the context of hierarchical and compositional semantics. This work was performed on artificial task to solve arthematic problem solving. (Giulianelli et al., 2018) has used probing on subject verb agreement task by probing LSTM layers. (Cohen et al., 2018) has applied probing in information retrieval. Probing has been profusely applied to analyse the learning capabilities of deep-contextualized embedding. (Lin et al., 2019; Clark et al., 2019; Tenney et al., 2019; Yu and Ettinger, 2020; Peters et al., 2020) has applied probing to analyse different linguistic features. (Lin et al., 2019) has applied it to analyse syntactical feature. (Clark et al., 2019) has applied probing on top of BERT's attention weights to analyse syntactic relation. They found that BERT can able to learn syntactical features, even if it is trained in unsupervised manner. (Peters et al., 2020) applied probing on ELMO embedding and showed it can learn hierarchy of contextual information like lower layer representation performed better in POS tagging task and higher layer in correference resolution. In recent days, some of the new publication has applied it to behavioural explanation, phrasal representation and composition, conversational recommendation and to check the understanding of idioms (Yu and Ettinger, 2020; Elazar et al., 2021b; Tan and Jiang, 2021; Penha and Hauff, 2020).

With the pervasive use of Probing in NLP, there is a word of caution came from (Hewitt and Liang, 2020; Belinkov, 2021). (Hewitt and Liang, 2020) has pointed out that a good score on a particular NLP task may not provide the true picture. The performance may be due to the learning capabilities of the probe itself. Such problems were acknowledged by (Zhang and Bowman, 2019) but a comprehensive analysis was put forth by Hewitt et. al. They have stressed on the fact that a good probe should have a good selectivity. To overcome the shortcomings, a concept of control task is proposed. Other authors have extended this concept in their works (Ravichander et al., 2021; Pimentel et al., 2020).

### 4.4 Natural language explanation

Methods which are presented in this survey or otherwise is not suitable for a layman. It is meant to be used by ML practitioners. So, it is imperative to use methods that can generate explanations for a layman person. It means the explanation generated by interpretability methods can be presented in a simple language or may be as a summary. Natural language explanation (NLE) has already been applied in the computer vision domain by . It was applied for the task like self-driving cars(Kim et al.), visual question answering task (Park et al.) and algebraic equation solving task by (Ling et al., 2017). This method has been applied in the NLP area as well. (Tim et al., 2018) proposed a two step process a two step process — "explain first then predict (reasoning)" and "predict first then explain (rationalization)". They argued that "explain first then predict" is more intuitive than the latter one. (Kumar and Talukdar, 2020) has proposed a model called NILE which follows the "explain first then predict" strategy to derive NLE. NILE generates multiple explantion, one for each label, the predict the answer based on the explanation. (Mccann et al., 2019) has proposed a model called as CAGE to generate an explanation for commonsense question answering. Language model inside the CAGE is based on GPT-2, a transformer (Vaswani, 2017) based architecture. NLE methods that are presented in this survey broadly falls in the local explanation category. These methods generates an explanation for a single instance. Model proposed by (Tim et al., 2018; Kim et al.; Ling et al., 2017; Mccann et al., 2019) falls in post-hoc category because the explanation is generated after the model is trained. However, NILE can be categorized to inherently interpretable method because it first generates the explanation.

### 4.5 Counterfactual explanations (CF) and Adversarial examples (AE)

Machine learning models tries to learn the correlation between the features and labels. Any statistical correlation is acceptable in ML framework, without considering the causality of those features. With the NLP applications are deployed in real world scenario, it is imperative to examined from the causality aspect to unearth the understanding of the model in the alternate scenarios (Moraffah et al., 2020; Feder et al., 2021a). (Moraffah et al., 2020) has pointed out to the three levels of interpretability listed by (Pearl, 2018) — Statistical interpretability, Causal interventional interpretability (Answers the question “What if") and Causal interpretability (Answers the question “Why?"). Rest of the survey has focused around the methods related to Statistical interpretability. In this section, this survey will focus on the Causal interpretability.

Causal interpretability in the NLP are mainly centered around counterfactual explanation (CE) and adversarial examples (AE). There is an ongoing debate around whether CEs and AEs are similar or different. This survey will briefly present that aspect followed by the employed methods in NLP area.

CEs and AEs are essentially a solution to the same optimization problem equation-3

$$\underset{x' \in \mathcal{X}}{\operatorname{argmax}}\, d(x, x') + \lambda d'(f(x'), y_des) \qquad (3)$$

Where $x$ is the original input and $x'$ is the CE/AE vector, $f$ is the model,$y_{des}$ is the output, $d$ and $d'$ is the distance, $\lambda$ is the trade-off. parameter (Freiesleben, 2021).

As pointed out by (Freiesleben, 2021), (Wachter et al., 2018) is of opinion that CE and AE are similar in nature but, differing in terms of the objective and in terms of data-points. This view is later countered by (Browne and Swift, 2020) pointing that AE, in all practical scenarios, remain very similar to the real-world input in order to have the imperceptibility. They held the view that they differ in terms of their semantic properties. However, (Verma et al., 2021) held that they are not same because their desiderata are different. Finally, (Freiesleben, 2021) has tried to put an unified framework for AEs and CEs. Where he differentiated AE and CE on the basis of — “ In relation to the true instance label and the constraint of how close the respective data point must be".

### 4.5.1 Adversarial examples

Evaluation of a model using adversarial examples are more centered towards robustness of the model. By useing AE, one can know the scenario in which its model is going to generate an incorrect output. It will provide an explanation that which type of edit has lead to the change in the output. In order to secure the model from AE attacks, models can be trained on adversarial data. (Ebrahimi et al., 2018) has proposed Hot-flip model to generate adversarial examples by flipping the character token. They have further suggested the method to generate AE by word level exchanges. To achieve this they have suggested there must be constraints like similarity, POS preservation etc, so that semantic should not be altered. They have applied in Text classification task. (Ribeiro et al., 2018b) has suggested a method called as Semantically Equivalent Adversarial Rules for AE generation. His method also stressed on the point that the AE should preserve the semantic equivalence. This method is applied in different task like Machine translation, VQA, Sentiment Analysis. (Hossam et al., 2020) has trained a white box interpretable substitute model to generate AE. (Sato et al., 2018) has proposed a method to perturb the word embedding to generate AE. Perturbations are guided towards existing word in the word embedding space. It will ensure that the resultant can be easily interpreted at sentence level. They have applied generated examples for Sentiment classification, grammatical error detection, category classification.

### 4.5.2 Counterfactual explanations

Similar to AE, a simple approach to explanation would be the generation of CE and compare the response of the model for normal input and counterfactual. Using CE, we can estimate the causal effect (Ross et al., 2021b; Gardner et al., 2020). CE generation can be achieve in two ways — manually and automatically. Manually writing CE for each of the input would be costly while generating it automatically may produce inconsistent counterfactuals.

Several authors have proposed a solution to this by altering the representations of the text in place of text itself. (Wu et al., 2021a) has proposed a framework called Polyjuice to create CE. It is domain agnostic in nature. It takes normal input sentence or masked ([BLANK]) input sentence with control command like negation, quantifier to generate CE. Generation of CE is done by transformer based language model GPT-2 (Radford et al., 2019). (Ross et al., 2021a) has proposed a CE generator model called MiCE. It is based on T5 (Raffel et al., 2020) and uses another form of counterfactual called *Contrastive Explanations*. T5 is fine tuned with input sentence and gold labels for the specified task. During counterfactual generation, it takes masked input and inverted label as an input. The amount of masked token is found by binary search and beam search is employed to keep track of the tokens which has altered the results with highest confidence. (Jacovi et al., 2021) has also employed *Contrastive Explanations* for interpretability. They have employed the methods similar to (Elazar et al., 2021a).

(Feder et al., 2021b) compute the counterfactual representation by pre-training an additional instance of the language representation model employed by the classifier, with an adversarial component designed to “forget" the concept of choice, while controlling for confounding concepts. (Elazar et al., 2021a) has used the concept of probing to generate CE. Principle aim of this method is to develop a model which can take neural representation as an input and produces an output that devoid of a specific information. They have iteratively trained an auxiliary classifier (as the case with probing) and projecting the representations into their null-space.

(Vig et al., 2020; Finlayson et al., 2021) has used causal mediation analysis. Mediation analysis relies on measuring the change in an output following a counterfactual intervention in an intermediate variable. It assumes Neural model as a graphical model from input to output with neurons as individual components.

## 5 Observation

This survey has broadly classified all the approaches into five broader categories. Further, it has highlighted the simple differences between causality-based and non-causal models. Causal methods, discussed in section – 4.5, has some relevance to input perturbation methods discussed in section – 4.1.2. In causal methods, the input features are perturbed to generate a different output from the model. (Rathi, 2019) has used SHAP(4.1.3) to produce counterfactual explanations. MiCE and PolyJuice use approach similar to gradient based method (4.1.1) to generate counterfactual examples.

We also observed that there is a lack of a common quantitative measure to measure interpretability. A simple evaluation approach relies on some form of *decrease in performance* of the model. Human-in-loop evaluation techniques are often employed. As pointed out by (Madsen et al., 2021), following the measures of interpretability (Doshi-Velez and Kim, 2017b) there exist some standard measures to measure interpretability. Such measures should be applied to build a unified approach.

Further, we would like to point out that some of the model's explanations are meant for Machine learning practitioners. Compared to those methods, Methods like Natural language explanations(4.4), Counterfactual explanations (4.5.2) and Adversarial examples (4.5.1) produces explanation in a simple language.

## 6 Conclusion

This survey has presented an overview of interpretability methods from a causal and non-causal perspective. In this survey, we have presented a brief overview of the different approaches and some theoretical discussion around those methods. We have presented the representative paper examples along with the specific NLP tasks they want to highlight.

## References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–13.

David Alvarez-Melis and T. Jaakkola. 2018. On the robustness of interpretability methods. *ArXiv*, abs/1806.08049.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. *CoRR*, abs/1606.07298.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Hosahalli Lakshmaiah Shashirekha. 2021. Fake news spreaders profiling using n-grams of various types and shap-based feature selection. *Journal of Intelligent & Fuzzy Systems*.

Yonatan Belinkov. 2021. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, (July):1–13.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:861–872.

Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *ICANN*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kieran Browne and Ben Swift. 2020. Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks.

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 *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. pages 276–286.

Daniel Cohen, Brendan O'Connor, and W. Bruce Croft. 2018. Understanding the representational power of neural retrieval models using NLP tasks. *ICTIR 2018 - Proceedings of the 2018 ACM SIGIR International Conference on the Theory of Information Retrieval*, (1):67–74.

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. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2126–2136.

Nelson Cowan. 2010. The magical mystery four: How is working memory capacity limited, and why? *Current Directions in Psychological Science*, 19(1):51–57. PMID: 20445769.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

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

Finale Doshi-Velez and Been Kim. 2017b. Towards A Rigorous Science of Interpretable Machine Learning. (Ml):1–13.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for NLP. *CoRR*, abs/1712.06751.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip : White-Box Adversarial Examples for Text Classification. pages 31–36.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021a. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021b. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2021a. Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021b. Causalm: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 1828–1843.

Samuel G. Finlayson, Isaac S. Kohane, and Andrew L. Beam. 2018. Adversarial attacks against medical deep learning systems. *CoRR*, abs/1804.05296.

Timo Freiesleben. 2021. The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples. *Minds and Machines*, pages 0–3.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna 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. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, pages 1307–1323.

Sahar Ghannay, Benoit Favre, Yannick Estève, and Nathalie Camelin. 2016. Word embedding evaluation and combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 300–305, Portorož, Slovenia. European Language Resources Association (ELRA).

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, (September):12–21.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards understanding neural machine translation with word importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 953–962, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2020. Designing and interpreting probes with control tasks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 2733–2743.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4129–4138.

Milo Honegger. 2018. Shedding light on black box machine learning algorithms: Development of an axiomatic framework to assess the quality of methods that explain individual predictions. *ArXiv*, abs/1808.05054.

Mahmoud Hossam, Trung Le, He Zhao, and Dinh Phung. 2020. Explain2Attack: Text adversarial attacks via cross-domain interpretability. In *Proceedings - International Conference on Pattern Recognition*, pages 8922–8928.

Dieuwke Hupkes and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:5617–5621.

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*, pages 4198–4205, Online. Association for Computational Linguistics.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive Explanations for Model Interpretability.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *NAACL*.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual Explanations for Self-Driving Vehicles. (i).

Arne Köhn. 2016. Evaluating Embeddings using Syntax-based Classification Tasks as a Proxy for Parser Performance. pages 67–71.

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural Language Inference with Faithful Natural Language Explanations. pages 8730–8742.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 681–691.

Yitong Li, Dianqi Li, Sushant Prakash, and Peng Wang. 2020. Toward interpretability of dual-encoder models for dialogue response suggestions. *ArXiv*, abs/2003.04998.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting inside BERT's Linguistic Knowledge. pages 241–253.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 158–167.

Zachary Chase Lipton. 2016. The mythos of model interpretability. *CoRR*, abs/1606.03490.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 289–297, Red Hook, NY, USA. Curran Associates Inc.

Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *ArXiv*, abs/1802.03888.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *ArXiv*, abs/1705.07874.

Ling Luo, Xiang Ao, Feiyang Pan, Jin Wang, Tong Zhao, Ningzi Yu, and Qing He. 2018. Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4244–4250. International Joint Conferences on Artificial Intelligence Organization.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc Interpretability for Neural NLP: A Survey.

Qianren Mao, Jianxin Li, Senzhang Wang, Yuanning Zhang, Hao Peng, Min He, and Lihong Wang. 2019. Aspect-based sentiment classification with attentive neural turing machines. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5139–5145. International Joint Conferences on Artificial Intelligence Organization.

David Mareek, Hande Celikkanat, Miikka Silfverberg, Vinit Ravishankar, and Jrg Tiedemann. 2020. Are multilingual neural machine translation models better at capturing linguistic features? *Prague Bull. Math. Linguistics*, 115:143–162.

Bryan Mccann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself ! Leveraging Language Models for Commonsense Reasoning. pages 4932–4942.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Christoph Molnar. 2019. *Interpretable Machine Learning*.

Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. 2020. Causal Interpretability for Machine Learning - Problems, Methods and Evaluation. *ACM SIGKDD Explorations Newsletter*, 22(1):18–33.

Pramod Kaushik 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 (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal Explanations : Justifying Decisions and Pointing to the Evidence.

Sungjoon Park, JinYeong Bak, and Alice H. Oh. 2017. Rotated word vector representations and their interpretability. In *EMNLP*.

Judea Pearl. 2018. Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. pages 3–3.

Gustavo Penha and Claudia Hauff. 2020. What does BERT know about books, movies and music? Probing BERT for Conversational Recommendation. *RecSys 2020 - 14th ACM Conference on Recommender Systems*, pages 388–397.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen Tau Yih. 2020. Dissecting contextual word embeddings: Architecture and representation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 1499–1509.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin van Durme. 2020. Collecting diverse natural language inference problems for sentence representation evaluation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 67–81.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary Chase Lipton. 2020. Learning to deceive with attention-based explanations. In *ACL*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam M. 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. *ArXiv*, abs/1910.10683.

Alessandro Raganato. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. pages 287–297.

Shubham Rathi. 2019. Generating Counterfactual and Contrastive Explanations using SHAP.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016a. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016b. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018a. Anchors: High-precision model-agnostic explanations. In *AAAI*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018b. Semantically equivalent adversarial rules for debugging NLP models. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:856–865.

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*, 8:842–866.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021a. Explaining NLP Models via Minimal Contrastive Editing (MiCE). pages 3840–3852.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E. Peters, and Matt Gardner. 2021b. Tailor: Generating and Perturbing Text with Semantic Controls.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 2:726–730.

Cynthia Rudin. 2018. Please stop explaining black box models for high stakes decisions. *ArXiv*, abs/1811.10154.

Barbara Rychalska, Dominika Basaj, Anna Wróblewska, and Przemyslaw Biecek. 2018. How much should you ask? on the question structure in QA systems. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 319–321, Brussels, Belgium. Association for Computational Linguistics.

Gözde Gül Şahin, Clara Vania, Ilia Kuznetsov, and Iryna Gurevych. 2020. LINSPECTOR: Multilingual probing tasks for word representations. *Computational Linguistics*, 46(2):335–385.

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:4323–4330.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? *ArXiv*, abs/1906.03731.

Lloyd Shapley. 1953. A value fo n-person games. *Ann. Math. Study28, Contributions to the Theory of Games, ed. by HW Kuhn, and AW Tucker*, pages 307–317.

Ying Shen, Yang Deng, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. 2018. Knowledge-aware attentive neural network for ranking question answer pairs. In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '18, page 901–904, New York, NY, USA. Association for Computing Machinery.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, (Table 2):1526–1534.

Jamin Shin, Andrea Madotto, and Pascale Fung. 2018. Interpreting word embeddings with eigenvector analysis.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *ICML*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

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*, AIES '20, page 180–186, New York, NY, USA. Association for Computing Machinery.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. 2020. Probing for referential information in language models. In *Proceedings* *of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4177–4189, Online. Association for Computational Linguistics.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2015. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Alona Sydorova, Nina Poerner, and Benjamin Roth. 2019a. Interpretable question answering on knowledge bases and text. In *ACL*.

Alona Sydorova, Nina Poerner, and Benjamin Roth. 2019b. Interpretable question answering on knowledge bases and text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4943–4951, Florence, Italy. Association for Computational Linguistics.

Hui Fen Tan, Kuangyan Song, Madeilene Udell, Yiming Sun, and Yujia Zhang. 2019. Why should you trust my interpretation? understanding uncertainty in lime predictions. *ArXiv*, abs/1904.12991.

Minghuan Tan and Jing Jiang. 2021. Does BERT Understand Idioms? A Probing-Based Empirical Study of BERT Encodings of Idioms. pages 1397–1407.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2021. Understanding Pure Character-Based Neural Machine Translation: The Case of Translating Finnish into English. pages 4251–4262.

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 ? *Iclr*, pages 1–17.

Oana-maria Camburu Tim, Rocktäschel Thomas, and Phil Blunsom. 2018. e-SNLI : Natural Language Inference with Natural Language Explanations. (NeurIPS):1–11.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9073–9080.

Akira Utsumi. 2020. Exploring What Is Encoded in Distributional Word Vectors: A Neurobiologically Motivated Analysis. *Cognitive Science*, 44(6):1–5.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *ArXiv*, abs/1909.11218.

Ashish Vaswani. 2017. Attention Is All You Need. (Nips).

Sahil Verma, John Dickerson, and Keegan Hines. 2021. Counterfactual Explanations for Machine Learning: Challenges Revisited.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias. 2(NeurIPS):1–14.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 183–196.

Sandra Wachter, Brent Mittelstadt, Chris Russell, I I Ntroduction, I I C Ounterfactuals, A Lsat Dataset, and B Pima Diabetes Database. 2018. Harvard Journal of Law & Technology Volume 31 , Number 2 Spring 2018 C OUNTERFACTUAL E XPLANATIONS W ITHOUT O PENING THE B LACK B OX : A UTOMATED D ECISIONS AND THE GDPR Harvard Journal of Law & Technology. 31(2).

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of nlp models is manipulable. *ArXiv*, abs/2010.05419.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. 2018. A game-based approximate verification of deep neural networks with provable guarantees. *CoRR*, abs/1807.03571.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021a. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 6707–6723.

Tongshuang (Sherry) Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021b. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *ACL/IJCNLP*.

Xiaoyan Yan, Fanghong Jian, and Bo Sun. 2021. Sakgbert: Enabling language representation with knowledge graphs for chinese sentiment analysis. *IEEE Access*, 9:101695–101701.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 4896–4907.

Muhammad Rehman Zafar and Naimul Mefraz Khan. 2019. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *ArXiv*, abs/1906.10263.

Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. 2010. Deconvolutional networks. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535.

Kelly Zhang and Samuel Bowman. 2019. Language Modeling Teaches You More than Translation Does: Lessons Learned Through Auxiliary Syntactic Task Analysis. pages 359–361.

Yu Zhang, Peter Tiño, Ales Leonardis, and Ke Tang. 2020. A survey on neural network interpretability. *CoRR*, abs/2012.14261.

Wei Zhao, Tarun Joshi, Vijayan N. Nair, and A. Sudjianto. 2020. Shap values for explaining cnn-based text classification models. *ArXiv*, abs/2008.11825.

Zhengze Zhou, Giles Hooker, and Fei Wang. 2021. S-lime: Stabilized-lime for model explanation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21, page 2429–2438, New York, NY, USA. Association for Computing Machinery.