# Analysis of Shortcut Learning Features in Natural Language Processing

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

**Wan Li**

Spring, 2022

Technical Project Team Members
Hanjie Chen
Andrew Wang

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

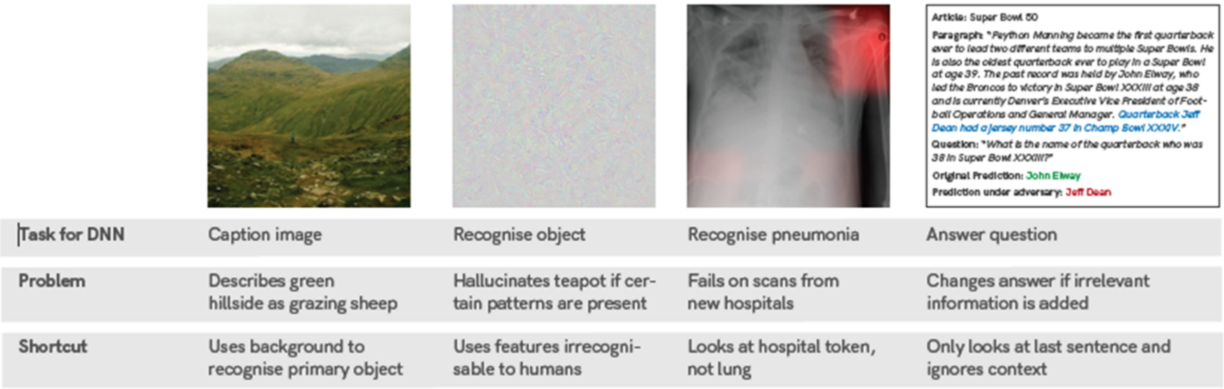
Yangfeng Ji, Department of Computer Science

# Analysis of Shortcut Learning Features in Natural Language Processing

**Wan Li**
**Department of Computer Science, University of Virginia, Charlottesville, VA, USA**
**wl9wgc@virginia.edu**

## Introduction

Many problems related to difficult machine learning problems are symptoms of shortcut learning. Shortcut learning, as Geirhos et al. puts it, are “decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions, such as real-world scenarios” (2020). In this paper, shortcuts are more specifically identified to be decision rules that perform well on in-distribution datasets, but lower performance in out-of-distribution datasets. At its core, shortcuts reveal a mismatch between the model’s intended solution and the learned solution; they are decision rules that do not align with the logic the model should have learned.



**Article:** Super Bowl 50

**Paragraph:** *“Peython Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had a jersey number 37 in Champ Bowl XXXIV.”*

**Question:** *“What is the name of the quarterback who was 38 in Super Bowl XXXIII?”*

**Original Prediction:** John Elway

**Prediction under adversary:** Jeff Dean

| Task for DNN | Caption image | Recognise object | Recognise pneumonia | Answer question |
|---|---|---|---|---|
| **Problem** | Describes green hillside as grazing sheep | Hallucinates teapot if certain patterns are present | Fails on scans from new hospitals | Changes answer if irrelevant information is added |
| **Shortcut** | Uses background to recognise primary object | Uses features irrecognisable to humans | Looks at hospital token, not lung | Only looks at last sentence and ignores context |

Figure 1: Examples of deep neural networks relying on shortcut features instead of learning the intended solution. Figure taken from Geirhos et al. (2020).

As shown in Figure 1, examples of shortcuts can include using a grassy background to identify the presence of sheep, or using what hospital a scan was taken from to determine if a patient has pneumonia instead of looking at the lung scan itself. Utilizing the wrong logic can lead to incorrect results when testing on data outside of the original testing set. These flaws in logic are extremely detrimental in real-life applications of machine learning technology; if a wrong conclusion is made with regards to cancer screening tests, autonomous vehicle driving, or job applications, the technology can be life-threatening.

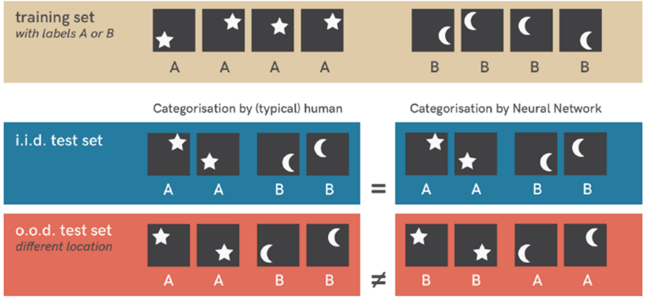


Figure 2: Toy example of a neural network learning to classify a dataset of stars and moons using the object's location instead of the object's shape. Figure taken from Geirhos et al. (2020).

Shortcuts are often difficult to identify. In the star and moon example in Figure 2, there are three different possible decision rules. The model could classify each image using the object's shape, location, or number of white pixels. Classification by shape is the intended solution, while the location and number of white pixels are shortcuts. In the field of natural language processing (NLP), a model can rely on a word's frequency in the training set to determine whether the input text has a positive or negative connotation, or it could analyze the last sentence of a paragraph to make a decision instead of taking into account the whole input (Du et al., 2021). These models can produce high accuracy results, but the model itself would use the wrong features for predictions, resulting in flawed logic connecting inputs to outputs (Geirhos et al., 2020). As a result, these models would perform poorly on out-of-distribution datasets. Identifying shortcuts in a model will allow researchers to develop better datasets and create machine learning models that learn the intended solution instead of relying on superficial correlations.

## Related Works

There have been many studies published recently regarding shortcut identification and mitigation, with the most notable being Geirhos et al. and Du et al.

Geirhos et al. provides the definition of a shortcut and provides a tier list of decision rules consisting of 1) all possible decision rules including non-solutions, 2) training solutions including overfitting solutions that work well on the training set but not on in-distribution data sets, 3) in-distribution test solutions, including shortcuts, that solves the training and in-distribution test sets, and 4) the intended solution that uses the intended features and performs well on out-of-distribution datasets (2020).

Du et al. identifies the first type of bias in NLP: lexical bias (2020). The researchers found that NLP models have a strong preference for features located at the head of the distribution, or features that are introduced first or most frequently (Du et al., 2021). This preference results in a lack of attention to important information that may be at the tail of the distribution, and the model may create shortcuts related to the features at the head. The paper also identifies two types of biases in machine learning models: data bias, in which the model learns superficial correlations due to the training set's data distribution, and model bias, where different models will have different outcomes given the same training data. The researchers proposed a mitigation framework that regularizes the distribution of features and suppresses the model from making overconfident predictions when the shortcut is deemed highly apparent (Du et al., 2021).

A second type of bias in NLP is the overlap bias, in which high word similarity between the premise and the hypothesis lead to predictions of entailment and low similarity results in contradiction (McCoy et al., 2019). A third type of bias present in NLP is the negation bias. Poliak et al. found that negation in the hypothesis side is strongly correlated with the "contradiction" label in SNLI, whereas Schuster et al. found that negation in the claim side is strongly correlated with with "refutes" label in FEVER (Poliak et al., 2018; Schuster et al., 2019).

A case study centered around visual commonsense reasoning found that question answering models, or models that try to determine the right answer choice to a question, do not perform as intended because they often cheat by using the frequency of words in the answers, or look at how many words the question and each answer choice have in common (Ye & Kovashka, 2021). The researchers approached the problem with an iterative masking technique, which ran the model several times while deleting, or masking, a different part of the input each time to determine which words are important and which words or phrases the model needs to ignore while making predictions.

Yang and Wang introduced a framework that automatically identifies shortcuts using interpretations on the model's behavior (2019). The framework consists of a cross-dataset analysis that identifies tokens that are more likely to be genuine than spurious, and knowledge-aware perturbation that checks how stable the prediction is by perturbing the extracted token to semantically similar neighbors. A weakness in Yang and Wang's paper includes the method's reliance on attention-based interpretation scores and its lack of dataset analysis (2019).

## Methods

### Setup

Two models were used in the experimental setup: BERT-base and RoBERTa-base (Liu et al., 2019; Tenney et al., 2019). Three different tasks were considered for each model, with an in-distribution and out-of-distribution dataset for each task: classification used IMDB and Yelp, natural language inference used SNLI and MNLI, and paraphrase identification used QQP and TwitterPPDB for the task's in-distribution and out-of-distribution dataset, respectively.

### Interpretation Methods

The accuracy of the model in terms of learning the intended solution can be measured using a model's interpretation. Interpretations shed light onto what logic a model uses to solve a problem. When the interpretation matches the intended solution, then the model has "learned" correctly and can solve the problem with the intended logic. Without interpretations, the model is a black box: the user can only see the inputs and outputs of the model, but there is no information about how the label is assigned. With the help of interpretation methods to explain a model's decisions, the technical project researchers can determine the correctness of a model and try to identify shortcuts present in the model.

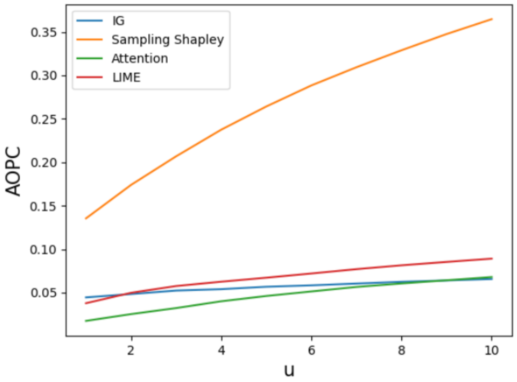


Figure 3. APOC scores of integrated gradients, sampling shapley, attention-based methods, and local interpretable model-agnostic explanations on the IMDB dataset using the BERT-base model.

The interpretation methods considered in this paper's experiments include integrated gradients (IG), sampling shapley, attention-based methods, and local interpretable model-agnostic explanations (LIME) (Castro et al., 2009; Pruthi et al., 2020; Ribeiro et al., 2016; Sundararajan et al., 2017). The four interpretation methods were run using the BERT-base model on the IMDB dataset. Results showed that sampling shapley is the best method, but it takes a very long time to run. As a result, sampling shapley was initially used for experiments but the interpretation method was later switched to attention-based methods for efficiency.

## Experiments

The project used interpretation methods to analyze model behavior and try to identify shortcut features from model explanations and data statistics using local mutual information (LMI). LMI focuses on the correlation between a feature and a label and can be calculated from a model's explanations. Further experiments tried to invoke randomness to see if variance in LMI would help identify shortcut features, which would be more likely to differ in importance between runs of different randomness.

The experiments initially started with the pretrained models BERT and RoBERTa. The models were each fine-tuned using the three in-distribution datasets (IMDB, SNLI, QQP) and three different random seeds per dataset (42, 400, 4000) to produce 18 different models. Each model was tested on in-distribution and out-of-distribution datasets to produce the model's accuracy, and then explanations were generated using the sampling  shapley interpretation method. LMI statistics were then generated using the three different random seed models for each dataset/model combination and the results were analyzed. Further experiments used different model ensembles, training dynamics, and data bootstrapping with an attention-based interpretation method to try to identify shortcut features.

## Results

Results from the experiments consisting of the BERT and RoBERTa model on the in-distribution and out-of-distribution datasets for the three NLP tasks with three different random seeds are found in Figure 4. There is little difference between the testing accuracy for different random seeds with the same model and dataset. However, there is a significant accuracy drop from testing a model on the in-distribution dataset versus the out-of-distribution dataset for each task except classification.

| Testing Accuracy | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task** | **Dataset** | **Seed 42, Run 1** | | **Seed 400, Run 2** | | **Seed 4000, Run 3** | |
| | | **BERT** | **RoBERTa** | **BERT** | **RoBERTa** | **BERT** | **RoBERTa** |
| **Classification** | **IMDB** | 91.571 | 93.572 | 91.35115664772272 | 93.1241495237333 | 91.62330905306972 | 93.22820779636596 |
| | **Yelp** | 90.989 | 93.103 | 91.3 | 91.92368421052632 | 90.31315789473683 | 93.5078947368421 |
| **Natural Language Inference** | **SNLI** | 90.205 | 91.079 | 90.46941678520626 | 91.05872790083316 | 90.12395854501118 | 90.97744360902256 |
| | **MNLI** | 74.485 | 77.155 | 73.20154880782556 | 78.21479519054412 | 73.83329936824944 | 76.68636641532504 |
| **Paraphrase Identification** | **QQP** | 91.422 | 91.457 | 91.22433836260203 | 91.62008409596834 | 91.23917882760327 | 91.3232747959436 |
| | **TwitterPPDB** | 87.782 | 86.943 | 87.76080877608088 | 87.99741879974188 | 87.37362873736288 | 86.87889868788987 |

Figure 4. Testing accuracy results from the different experimental models. For each task, there are two datasets: the top dataset is the in-distribution dataset and the bottom dataset is the out-of-distribution dataset. Each model (BERT, RoBERTa) was run three times using different random seeds (42, 400, 4000) on each dataset.

Each model produced explanations using the interpretation methods, and the explanations were used to calculate LMI scores. Each model can produce its own LMI score for each word token, but the LMI mean and variance for each token were found using multiple models with different random seeds per model and dataset. In Figure 5, the x-axis represents the LMI mean and the y-axis represents the LMI variance. Each point on the cartography represents a token in the input.

Model ensembles

ep=3, seeds: 42, 400, 4000, 40000, 400000 Neg

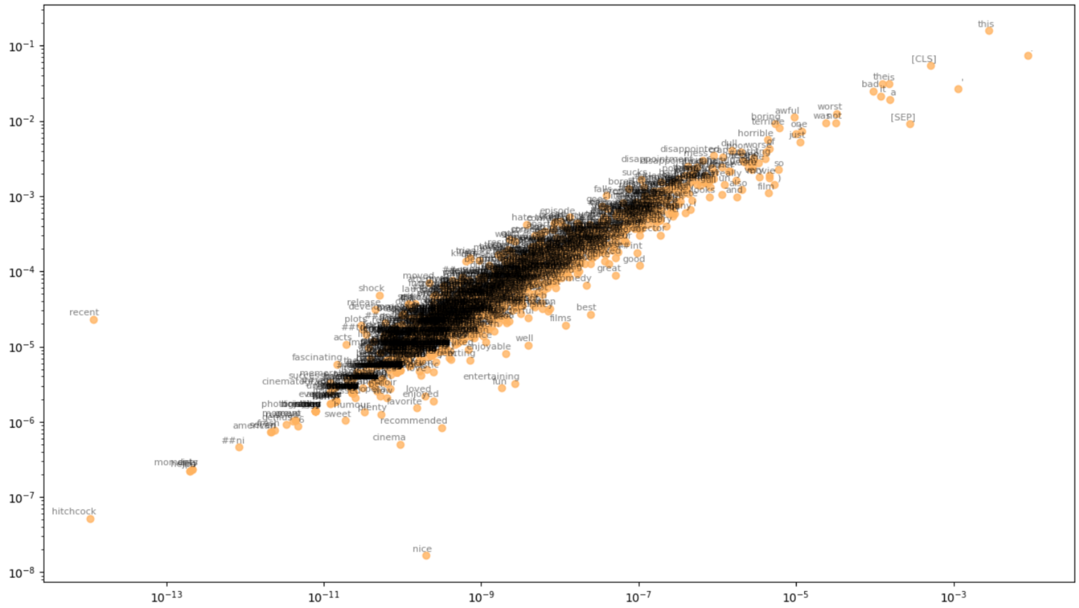


Figure 5. Token cartography based on LMI mean (x-axis) and variance (y-axis) for different model ensembles with various random seeds on the negative label. Tokens higher up in the graph have higher frequency.

We can distinguish between shortcuts and genuine tokens a little bit based on the word frequency in the in-distribution dataset and the LMI distribution, but there is not enough of a difference between genuine tokens and spurious tokens to tell. For example, spurious tokens like “the” and “it” have a similar score to the genuine token “bad”. In addition, many tokens towards the middle are overlapped, making the results hard to read. The model’s behavior is very similar between shortcut features and genuine tokens and it is difficult to differentiate the two. Thus, it was concluded that shortcuts cannot be identified using only data statistics and in-distribution data.

## Future Work

Out-of-distribution data can be analyzed to see if shortcuts can be identified with outside datasets. The same experimental process would be used, in which data statistics and LMI are gathered. However, different approaches can be used as necessary to obtain results.

Another next step includes human evaluation for identifying shortcuts. The model can be fine-tuned on an in-distribution dataset and then have a human annotate the results. The model can then be retrained with the human annotations and the experiment would observe if the model performs better than before and whether the amount of shortcuts would diminish.

Finally, once the shortcuts have been identified, a strategy can be created to prevent or mitigate shortcuts in NLP models. Mitigation strategies are the ultimate goal of shortcut learning research in order to understand and further improve machine learning models.

## Conclusion

Through experiments with multiple models, datasets, and model ensembles, it can be concluded that using only data statistics and in-distribution datasets is inadequate to identify shortcuts. Previous works that use only data statistics and in-distribution datasets to identify and mitigate shortcuts will need to reconsider their approaches, as there is not enough information to find shortcuts in the data from those sources alone. Additional information would be needed to accurately find the model's spurious tokens, which may include looking at out-of-distribution datasets. If shortcuts can be accurately identified and mitigated, then machine learning models can improve drastically and be more reliable in real-world situations.

# References

Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, *36*(5), 1726–1730. https://doi.org/10.1016/j.cor.2008.04.004

Du, M., Manjunatha, V., Jain, R., Deshpande, R., Dernoncourt, F., Gu, J., Sun, T., & Hu, X. (2021). Towards Interpreting and Mitigating Shortcut Learning Behavior of NLU models. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 915–929. https://doi.org/10.18653/v1/2021.naacl-main.71

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, *2*(11), 665–673. https://doi.org/10.1038/s42256-020-00257-z

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv:1907.11692 [Cs]*. http://arxiv.org/abs/1907.11692

McCoy, T., Pavlick, E., & Linzen, T. (2019). Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3428–3448. https://doi.org/10.18653/v1/P19-1334

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., & Van Durme, B. (2018). Hypothesis Only Baselines in Natural Language Inference. *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 180–191. https://doi.org/10.18653/v1/S18-2023

Pruthi, D., Gupta, M., Dhingra, B., Neubig, G., & Lipton, Z. C. (2020). Learning to Deceive with Attention-Based Explanations. *ArXiv:1909.07913 [Cs]*. http://arxiv.org/abs/1909.07913

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-Agnostic Interpretability of Machine

Learning. *ArXiv:1606.05386 [Cs, Stat]*. http://arxiv.org/abs/1606.05386

Schuster, T., Shah, D., Yeo, Y. J. S., Roberto Filizzola Ortiz, D., Santus, E., & Barzilay, R. (2019). Towards Debiasing Fact Verification Models. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3419–3425. https://doi.org/10.18653/v1/D19-1341

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. *ArXiv:1703.01365 [Cs]*. http://arxiv.org/abs/1703.01365

Tenney, I., Das, D., & Pavlick, E. (2019). BERT Rediscovers the Classical NLP Pipeline. *ArXiv:1905.05950 [Cs]*. http://arxiv.org/abs/1905.05950

Yang, Y., & Wang, X. (2019). Modeling the intention to use machine translation for student translators: An extension of Technology Acceptance Model. *Computers & Education*, *133*, 116–126. https://doi.org/10.1016/j.compedu.2019.01.015

Ye, K., & Kovashka, A. (2021). A Case Study of the Shortcut Effects in Visual Commonsense Reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(4), 3181–3189.