Interpreting Predictions of NLP Models

Eric Wallace
UC Berkeley
ericwallace@berkeley.edu

Matt Gardner
Allen Institute for AI
mattg@allenai.org

Sameer Singh
UC Irvine
sameer@uci.edu

Abstract

Although neural NLP models are highly expressive and empirically successful, they also systematically fail in counterintuitive ways and are opaque in their decision-making process. This tutorial will provide a background on interpretation techniques, i.e., methods for explaining the predictions of NLP models. We will first situate *example-specific* interpretations in the context of other ways to understand models (e.g., probing, dataset analyses). Next, we will present a thorough study of example-specific interpretations, including saliency maps, input perturbations (e.g., LIME, input reduction), adversarial attacks, and influence functions. Alongside these descriptions, we will walk through source code that creates and visualizes interpretations for a diverse set of NLP tasks. Finally, we will discuss open problems in the field, e.g., evaluating, extending, and improving interpretation methods. The tutorial slides and the accompanying code is available online at https://www.ericswallace.com/interpretability.

1 Tutorial Description

Neural models have become the de-facto standard tool for NLP tasks. These models are becoming increasingly powerful—recent work shows that large neural models substantially improve accuracy on a wide range of downstream tasks (Devlin et al., 2019; Brown et al., 2020). However, today's models still make egregious errors: they reinforce racial biases (Sap et al., 2019), fail in counterintuitive ways (Jia and Liang, 2017; Feng et al., 2018), and often solve tasks using simple surface-level patterns (Gururangan et al., 2018; Min et al., 2019).

These model insufficiencies are exacerbated by the inability to understand *why* models made the predictions they do. Interpretation methods seek to fill this void. In particular, *example-specific* interpretations provide post-hoc explanations for individual model predictions. These explanations come in various forms, e.g., attributing the importance of the input features through saliency maps (Smilkov et al., 2017), perturbing the inputs and observing the model's response (Feng et al., 2018; Ribeiro et al., 2018b), or locating a model's local decision boundary (Ribeiro et al., 2016).

This tutorial will provide an introduction to the various types of example-specific interpretations. We will present the technical details of existing methods, including saliency maps, adversarial attacks, input perturbations, influence functions, and other methods. We will cover how these interpretations are applied to various tasks and input-output formats, e.g., text classification using LSTMs, masked language modeling using BERT (Devlin et al., 2019), and text generation using GPT-2 (Radford et al., 2019).

For each task, we will walk through example use cases of interpretations: highlighting model weaknesses (Jia and Liang, 2017), increasing/decreasing user trust (Feng et al., 2018), and understanding hard-to-formalize criteria such as bias, safety, and fairness (Doshi-Velez and Kim, 2017). Alongside the tutorial, we will present source code implementations of various interpretation methods using AllenNLP Interpret (Wallace et al., 2019b).

2 Details and Prerequisites

The tutorial will be of the *cutting-edge* type. The tutorial slides and the accompanying code is available online at https://www.ericswallace.com/interpretability.

Prerequisites Attendees should have a basic understanding of different tasks in NLP such as text classification, sequence tagging, and reading comprehension (predicting spans in a passage).

Attendees should also have a basic understanding of neural network methods for NLP, including:

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts, pages 20–23
Online, November 19 - 20, 2020.
https://doi.org/10.18653/v1/P17

- How backpropagation can compute gradients with respect to the parameters.
- How tokens/words are represented (i.e., word and sub-word embeddings).
- High-level ideas behind different model architectures (e.g., RNNs, Transformers).
- Optional knowledge of contextualized embedding models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019).

Finally, a portion of the tutorial will walk through Python code samples in PyTorch and AllenNLP (Gardner et al., 2018b). Participants do not need to understand this code to follow the main tutorial material.

Reading List Doshi-Velez and Kim (2017) provide a great overview and motivation for interpretability research. Lipton (2018) and Jain and Wallace (2019) discuss some of the challenges of defining and evaluating interpretability. Jia and Liang (2017) help demonstrate the fragility of NLP models. LIME (Ribeiro et al., 2016) and saliency maps (Simonyan et al., 2014) are now standard interpretations. Wallace et al. (2019b) provides example NLP interpretations (interested readers can inspect their code).

3 Tutorial Outline

The tutorial will present three hours of content with a thirty-minute break.

Motivation This section will discuss why we care about interpretability. It will paint a landscape of today's neural models, describe how models are brittle and behave counterintuitively, and explain how interpretations can open the "black box" of machine learning.

Introduction to Interpretations This section will situate *example-specific* interpretations in the context of other methods. We will discuss:

- Dataset analyses, e.g., error analysis, Errudite (Wu et al., 2019), diagnostic "challenge" test sets (Naik et al., 2018; Gardner et al., 2020)
- "Probing", i.e., inspecting a model's embeddings for certain properties (Liu et al., 2019; Tenney et al., 2019).
- Rationale-based explanations, i.e., a model generates text for why it made its prediction.
- Example-specific interpretations (our tutorial's focus), e.g., saliency maps (Simonyan et al., 2014), LIME (Ribeiro et al., 2016), adversarial attacks (Szegedy et al., 2014), and input perturbations (Feng et al., 2018).

Example-specific Interpretations This section will introduce example-specific interpretations in more detail. We will discuss the challenges and approaches to evaluating such interpretations. We will also cover the critiques and shortcomings of using attention as explanations (Jain and Wallace, 2019; Serrano and Smith, 2019). We will then explain why we focus on *gradient-based methods*: they are model-agnostic, easy to compute, and (largely) faithful to a model's behavior.

Understanding What Parts of An *Input* Led to a Prediction This section will discuss:

- *Saliency maps*, i.e., generating visualizations of "salient" input tokens. We will discuss how to generate saliency maps using gradient-based techniques (Simonyan et al., 2014; Sundararajan et al., 2017; Smilkov et al., 2017)) and black-box techniques (Ribeiro et al., 2016).
- *Input Perturbations*, i.e., showing how changes to the input do (or do not) change the prediction. For example, leave-one-out (Li et al., 2016) and input reduction (Feng et al., 2018). We will also cover *adversarial* perturbations such as token flipping (Ebrahimi et al., 2018) and adding distractor sentences (Jia and Liang, 2017).

Break

Understanding How *Global* Decision Rules Led to a Prediction This section will discuss how certain global "decision rules" can explain model predictions. We will cover Anchors (Ribeiro et al., 2018a) and Universal Adversarial Triggers (Wallace et al., 2019a). We will also discuss how spurious patterns in datasets, e.g., lexical overlap in textual entailment (McCoy et al., 2019), can cause models to learn certain undesirable decision rules.

Understanding Which *Training Examples* Caused a Prediction This section will discuss how to trace model predictions back to the training data, i.e., identifying "influential" training points. We will cover influence functions (Koh and Liang, 2017) and representor points (Yeh et al., 2018).

Coding Interpretations This section will walk through source code for selected interpretation methods. Using AllenNLP Interpret (Wallace et al., 2019b), we will cover example use cases such as interpreting LSTM-based sentiment analysis models and BERT-based masked language models.

Open Problems We will conclude with a discussion of areas for future research:

- *Evaluation:* There is fundamentally no ground-truth to use for evaluating interpretations; how do we define evaluation?
- *Robustness & Faithfulness:* Interpretations may be unfaithful to the underlying model and can be adversarially manipulated. What are the implications of this, and how can we improve existing interpretation methods?
- *Interpretation Beyond Classification:* Most interpretations focus on classification models; how are interpretations best applied to the complex input-output formats seen in NLP tasks (e.g., machine translation)?
- *Closing the loop with Humans:* Humans are the end-users of interpretations; how can we make interpretations interactive, collaborative, customizable, and ultimately more effective?
- *Pretrained Transformer Models:* How do our methods, and the field of interpretability, change with the rise of massively-pretrained models?

4 Instructors

Eric Wallace is a PhD student at the University of California, Berkeley. His research focuses on the interpretability and robustness of machine learning models for NLP. He is the lead developer of the AllenNLP Interpret toolkit and has published numerous papers on interpreting neural NLP models. Website: http://ericswallace.com

Matt Gardner is a senior research scientist at the Allen Institute for Artificial Intelligence (AI2). His research focuses on question answering, semantic parsing, and model analysis. Matt received his PhD from the Language Technologies Institute at Carnegie Mellon University. He is the lead designer of the AllenNLP toolkit and a host of the NLP Highlights podcast.

Matt was an instructor at the Neural Semantic Parsing Tutorial (Gardner et al., 2018a) at ACL 2018, and the Writing Code for NLP Research Tutorial (Gardner et al., 2018c) at EMNLP 2018. Website: https://matt-gardner.github.io/

Sameer Singh is an Assistant Professor of Computer Science at the University of California, Irvine. He is working on large-scale and interpretable machine learning models for NLP. Before UCI, Sameer was a Postdoctoral Research Associate at the University of Washington, and he received his PhD from the University of Massachusetts, Amherst in 2014.

Sameer presented the Deep Adversarial Learning Tutorial (Wang et al., 2019) at NAACL 2019 and the Mining Knowledge Graphs from Text Tutorial at WSDM 2018 and AAAI 2017. Sameer has also received teaching awards at UCI. Website: http://sameersingh.org/

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

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

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *COLING*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *EMNLP*.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating NLP models via contrast sets. *arXiv preprint arXiv:2004.02709*.

Matt Gardner, Pradeep Dasigi, Srinivasan Iyer, Alane Suhr, and Luke Zettlemoyer. 2018a. Neural semantic parsing. In *ACL Tutorial*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018b. Allennlp: A deep semantic natural language processing platform. In *ACL Workshop for NLP Open Source Software*.

Matt Gardner, Mark Neumann, Joel Grus, and Nicholas Lourie. 2018c. Writing code for NLP research. In *EMNLP*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*.

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

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *ICML*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Zachary C Lipton. 2018. The mythos of model interpretability. *Queue*.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *NAACL*.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *COLING*.

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

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

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *KDD*.

Marco Túlio 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. In *ACL*.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *ACL*.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *ACL*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. SmoothGrad: removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *ICML*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

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

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP*.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019b. AllenNLP Interpret: A framework for explaining predictions of NLP models. In *EMNLP*.

William Yang Wang, Sameer Singh, and Jiwei Li. 2019. Deep adversarial learning for NLP. In *NAACL Tutorial*.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2019. Errudite: Scalable, reproducible, and testable error analysis. In *ACL*.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. 2018. Representer point selection for explaining deep neural networks. In *NeurIPS*.