The Design and Evaluation of Neural Attention Mechanisms for Explaining Text Classifiers

by

Samuel Carton

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information)
in The University of Michigan
2019

Doctoral Committee:

Professor Qiaozhu Mei, Co-chair
Professor Paul Resnick, Co-chair
Professor Eytan Adar
Professor Vinod Vydiswaran

Samuel Carton

scarton@umich.edu

ORCID iD: 0000-0001-7520-0400

ACKNOWLEDGEMENTS

My first and most fervent thanks go to my advisers Paul Resnick and Qiaozhu Mei. Without their patience, supportiveness, inventiveness and expertise, I would not have been able to navigate this process.

I thank my committee members Vinod Vydiswaran and Eytan Adar, as well as the many, many friends and colleagues who have supported, mentored and consoled me over my years at the University of Michigan. These include Daphne Chang, Chanda Phelan, Hari Subramonyam, Xin Rong, Wei Ai, Matt Burgess, Shiyan Yan, Yue Wang, Sangseok You, David Jurgens, Daniel Romero, Ceren Budak, Eric Gilbert, Ryan Burton, Heeryung Choi, Jackie Cohen, Ashwin Rajadesingan, and Souneil Park.

I thank the administrative staff at the School of Information, particularly Veronica Falandino, Rebecca Epstein, Kanda Fletcher, Allison Sweet, Rebecca O'Brien, and Barb Smith. UMSI is an amazingly healthy, friendly and well-run place, and it is the school's staff who make it so. I also acknowledge the Rackham Graduate School student records staff, particularly Arahshiel Silver and Sue Bathgate, who have been exceptionally helpful in the process of formally submitting this document.

Finally, I want to thank my family and my wonderful fiancée for their unflagging and unconditional support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ABSTRACT

The last several years have seen a surge of interest in interpretability in AI and machine learning–the idea of producing human-understandable explanations for AI model behavior. This interest has grown out of concerns about the robustness and accountability of AI-driven systems, particularly deep neural networks, in light of the increasing ubiquity of such systems in industry, science and government. The general hope of the field is that by producing explanations of model behavior for human consumption, one or more model-using stakeholder groups (e.g. model designers, model-advised decision-makers, recipients of model-driven decisions) will be able to derive some type of increased utility from those models (e.g. easier model debugging, better decision-making, higher user satisfaction).

The early years of this field have seen a profusion of technique but a paucity of evaluation. A number of methods have been proposed for explaining the decisions of deep neural models, or of constraining neural models to behave in more interpretable ways. However, it has proven difficult for the community to reach a consensus about how to evaluate the quality of such methods. Automated evaluation protocols such as collecting gold-standard explanations do not necessarily correlate well with true practical utility, while fully application-oriented evaluations are expensive, difficult to generalize from, and, it increasingly appears, an extremely difficult HCI challenge.

In this work I address gaps in both the design and evaluation of interpretability methods for text classifiers.

I present two novel interpretability methods. The first method is a feature-based explanation technique which uses an adversarial attention mechanism to identify all

predictive signal in the body of an input text, allowing it to outperform strong baselines with respect to human gold-standard annotations. The second method is an example-based technique that retrieves explanatory examples using only the features that were important to a given prediction, leading to examples which are much more relevant than those produced by strong baselines.

I accompany each method with a formal user study evaluating whether that type of explanation improves human performance in model-assisted decision-making. In neither study am I able to demonstrate an improvement in human performance as an effect of explanation presence. This, along with other recent results in the interpretability literature, begins to reveal an intriguing expectation gap between the enthusiasm that the interpretability topic has engendered in the machine learning community and the actual utility of these techniques in terms of human outcomes that the community has been able to demonstrate.

Both studies represent contributions to the design of evaluation studies for interpretable machine learning. The second study in particular is one of the first human evaluations of example-based explanations for neural text classifiers. Its outcome reveals several important, nonobvious design issues in example-based explanation systems which should helpfully inform future work on the topic.

CHAPTER I

Introduction

Interpretable machine learning, sometimes referred to as explainable machine learning or explainable AI (XAI), seeks to explain the predictions of machine learning models in human-understandable terms.

This is an important area of study. Recent years have seen a huge proliferation in applications of machine learning to every aspect of society including other areas of science such as medicine, business and government. At the same time, the rise of neural networks as the dominant machine learning paradigm has lead to predictive models that are unprecedentedly vast, powerful, complex and opaque in comparison to older modeling styles. While these recent improvements in modeling power have increased the applicability of machine learning, this opacity continues to limit its safety (*Guidotti et al.*, 2018; *Gilpin et al.*, 2018; *Murdoch et al.*, 2019).

Machine learning models make mistakes. They have a tendency to be brittle, having a hard time generalizing beyond whatever particular circumstances are present in the training data. They often end up absorbing bias from their training data. They are vulnerable, in some domains, to adversarial examples–inputs which have been manipulated in ways that are imperceptible to the human eye, but which result in vastly different model output (*Gilpin et al.*, 2018). Even beyond these kinds of pathological model behaviors, there are many tasks where it simply is not possible to

train a completely reliable model due to noisy or sparse data (e.g. deceptive review detection (*Lai and Tan*, 2019)), or intrinsic randomness in the outcome relative to the available signal (e.g. police officer misconduct (*Carton et al.*, 2016)). The opacity of complex models can often make it difficult to recognize when these types of mistakes have occurred.

Humans make mistakes. In many domains, models have been found to be more generally reliable than human experts and in some cases have been shown to make different mistakes from those made by humans (e.g *Kleinberg et al.* (2017); *Ardila et al.* (2019)). However, the crucial difference between model mistakes and human mistakes is that human decisions are generally (at least nominally) reason-driven. A judge who convicts a defendant does so based on evidence–if the evidence is shown to be false the conviction can be overturned. A physician accused of malpractice can try to justify their decisions for scrutiny by a third party.

This accountability is crucial because it allows bad or unfair decisions to be recognized, reported, appealed and reversed. If someone is denied a loan as a result of a bankruptcy they did not experience, an explanation of their denial will give them the information they need to dispute the decision where they might otherwise have simply given up. This type of affordance is the stated rationale behind the "right to explanation" clause of the recent European Union General Data Protection Regulation (GDPR):

> In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision. (*Wachter et al.*, 2017a)

Equally important to the ability to overturn bad decisions is the ability to learn from them. When John Hinckley Jr. attempted to assassinate US President Ronald

Reagan, the Insanity Defense Reform Act was passed to address what was perceived as a failure to sentence him appropriately (*Finkel*, 1989). More recently, when a diabetic woman in the United Kingdom lost her child because she opted for a natural birth whose risks she hadn't been made properly aware of, the subsequent court case set a new standard for obtaining informed consent from patients (*Whittaker*, 2015). In both of these cases, a precise understanding of the reasons behind the precipitating failure event were crucial in adjusting the "algorithm" involved to be more optimal.

What interpretability can bring to machine learning is accountability. Individual machine decisions may be more accurate on average than human decisions, but it is only through interpretability that those decisions can be incorporated into the kind of robust *decision ecosystem* described above, where a decision is a dynamic object capable of being critiqued, changed and learned from.

This quality of accountability is important across many different levels of human-machine cooperation. In scenarios like recidivism prediction or medical diagnosis, where every decision needs to be carefully weighed and considered, interpretability can give human operators a way to more easily integrate machine advice into their own decision processes. However, even in more automated scenarios such as fingerprint recognition, interpretability can give model builders a way to understand and debug the occasional model errors that their systems will inevitably produce.

The challenge for machine learning researchers and engineers is to design interpretability methods which afford a level of human understanding to make this kind of accountability possible. However, how exactly to operationalize the notion of "human understanding" is a topic of active discussion within the interpretable machine learning literature, as it cuts to the very heart of what is meant by "interpretable". As I discuss below, that literature has proposed a variety of potential answers to the question of "what is interpretability?", as well as a profusion of methods designed to achieve these proposed objectives.

1.1 Interpretable Machine Learning

1.1.1 Why do we want interpretability?

In the above introduction, I make the case for interpretability as a way of improving the **robustness** of AI-in-the-loop decision systems. However, the literature has suggested a number of reasons why interpretable models might be desirable. Three well-cited reviews (*Lipton et al.*, 2016; *Doshi-Velez and Kim*, 2017; *Samek et al.*, 2017) produce the following combined list of use cases for how the presence of explanations might increase the utility of machine learning models. I preserve the exact wording used by the original works in order to convey the range of ways these ideas have been articulated.

Learning from the model/causality/scientific understanding Explanations can suggest causal relationships in the domain that are of general interest to a human observer, and which can prompt further investigation (*Samek et al.*, 2017; *Lipton et al.*, 2016; *Doshi-Velez and Kim*, 2017).

Safety/verification Explanations may allow human overseers to better supervise the functioning of a model that is not reliable enough to be allowed to operate on its own (*Doshi-Velez and Kim*, 2017; *Samek et al.*, 2017).

Mismatched objectives/transferability Explanations may expose generalization issues arising from cases where a model is trained against objectives that do not entirely match their desired application (*Doshi-Velez and Kim*, 2017; *Lipton et al.*, 2016).

Informativeness Explanations may allow models to provide more useful advice to human overseers in making decisions (*Lipton et al.*, 2016).

Trust Explanations may increase user confidence in the model (*Lipton et al.*, 2016).

Ethics Explanations may reveal discrimination in the reasoning behind a model's predictions (*Doshi-Velez and Kim*, 2017).

Multi-objective trade-offs Explanations may clarify how two competing objectives in a model are being balanced against one another (*Doshi-Velez and Kim*, 2017).

Improvement of the system Explanations can provide clues about how to improve a faulty model (*Samek et al.*, 2017).

Compliance to legislation Explanations can enable compliance with laws like the GDPR right to explanation (*Samek et al.*, 2017).

1.1.2 How do we quantify interpretability?

Highly related to the question of why we want interpretability is how we measure it. What does it mean for an algorithm to be interpretable, or for an explanation of an algorithm's behavior to be a good explanation? Like the motivations listed above, the literature is far from settled on this question, but a number of desiderata have been identified.

Sparsity The most common desired quality of an explanation is sparsity–that an explanation should reduce the informational complexity of the underlying decision process by some significant degree. In the most common interpretability approach, feature attribution, this amounts to reducing a model input to just those features which had a particularly large impact on the model output (*Murdoch et al.*, 2019; *Doshi-Velez and Kim*, 2017).

Usability Another common desired quality of explanations is that they should increase the user satisfaction, trust, and overall usability of AI-powered systems, as might be measured by a survey (*Abdul et al.*, 2018).

Decision quality To directly assess the impact of explanations on the robustness of a system, we can measure the quality of the decisions that humans make in collaboration with an interpretable system (*Doshi-Velez and Kim*, 2017).

Simulatability Another common goal is that of simulatibility: that for an explanation to cause a human to "understand" a model is for them to be able to look at the explanation and simulate the behavior of the model by correctly guessing what it will predict (*Murdoch et al.*, 2019; *Doshi-Velez and Kim*, 2017).

Comparison to human-produced explanations Some authors use similarity to human-produced explanations as criteria for machine-generated explanations, for example showing that the machine attention used by a classifier matches the visual attention used by human subjects on an image recognition task, or that machine-produced explanations match human-produced explanations for the same task (e.g. *Lei et al.* (2016); *Mohseni and Ragan* (2018)).

Fidelity An often-discussed quality is the notion of fidelity–that an explanation should be faithful to the behavior of the underlying model (*Murdoch et al.*, 2019). Some interpretability techniques (e.g. (*Ribeiro et al.*, 2016)) are based on the idea of approximating the behavior of a complex model with a simple model, and then using the structure of the simple model to explain the behavior of the complex model. However, there are concerns that these types of explanations are really only "explaining" the approximation, and that they may in fact be misleading in cases where the approximation differs from the underlying model (*Rudin*, 2019).

Uniqueness A final quality that has undergone recent discussion is the idea of uniqueness–that a good explanation should have a one-to-one mapping with the model decision it is intended to explain. A good explanation method should not be able to produce multiple equally-plausible explanations for the same model output, nor should a given explanation be equally plausible for a range of model outputs (*Jain and Wallace*, 2019).

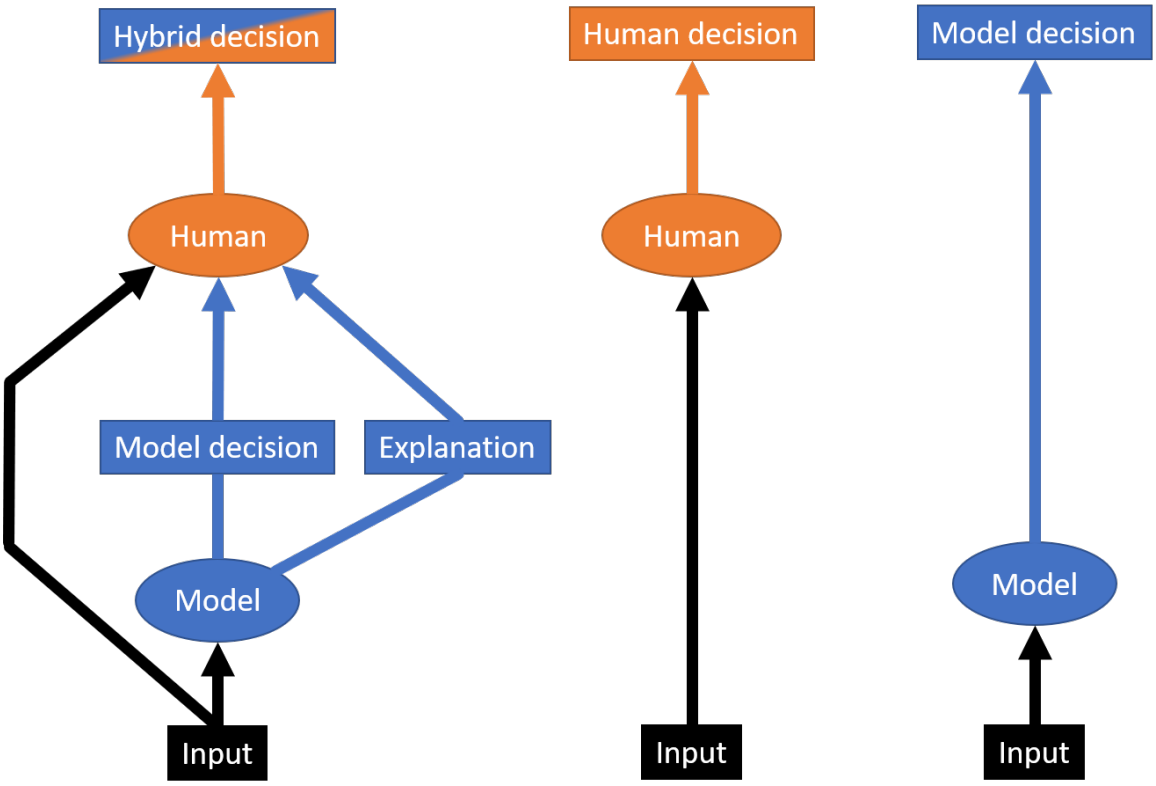

Figure 1.1: Diagram of how a human and model agent interact in a hybrid system (left) as opposed to either type of agent operating alone (right).

1.1.3 It's (almost) all about robustness

Explanations are intended for human consumption. What this means is that when we discuss interpretable machine learning, we are implicitly discussing a hybrid system (Figure 1.1) in which a trained model makes a prediction and produces an explanation

for the benefit of a human overseer, who then has to make a choice about whether they want to trust the model. This trust decision can happen on an immediate time scale (do I trust the model on this particular prediction?) or on a longer scale (do I generally trust this model enough to let it operate with autonomy?), but the ultimate goal is for good decisions to be made.

I argue that most of the above-listed motivations and desiderata for interpretability can be reduced to a hope that explanations can improve the robustness of this hybrid system. They reflect a concern that even very powerful modern models (i.e. deep neural networks) are not reliable enough to operate with full autonomy. They reflect a hope that interpretability can enable efficient human auditing of such models in order to ultimately improve the quality of decisions made in conjunction with them. The connection between this idea of "hybrid robustness" and accountability as discussed earlier is that accountability drives robustness–accountable decisions are decisions that can be audited in service to a more robust outcome.

Safety/verification and its direct operationalization of measuring decision quality are the clearest example of this, asking interpretability to allow humans to troubleshoot the decisions of a faulty model. However, transferability and ethics, while presented as separate concepts, both just represent specific ways for a model to be incorrect (by generalizing poorly or discriminating against certain groups respectively). The idea of informativeness merely shifts focus to the idea of an interpretable model improving the decisions of a human expert. In all cases, the ultimate goal is for the combined human-model system to make fewer mistakes with the benefit of interpretability than without.

Trust and usability can be articulated in terms of decision quality as well. A well-designed interpretability system *should* inspire user trust and satisfaction in a model–but only when that model is correct. They should prompt suspicion when the model is incorrect. That is, the trust that explanations engender should be properly

calibrated to the true reliability of the model, to the ultimate benefit of decisions made by human users in conjunction with that model.

With respect to legislative compliance, the motivation behind the GDPR "right to explanation" concept is to allow the subjects of algorithmic decision-making to audit the decisions that affect their lives. The implicit expectation is that these subjects should be able to recognize and appeal these decisions when they are based on flimsy or spurious reasoning. For example, an individual denied a loan by an automated risk assessment algorithm should in theory be able to recognize when the algorithm is utilizing an inaccurate accounting of their credit score and be able to appeal that decision.

Perhaps the hardest motivation to link to the idea of robustness is that of generating scientific insight, since this goal abstracts away from the idea of a specific decision task that can be performed with more or less accuracy. However, the mark of a strong scientific theory is that it has good predictive power–robustness, in fact. So even this use case for interpretable machine learning can be thought of as enabling hybrid robustness of a sort, it is simply that at some point under this scheme the actual model is removed from consideration, leaving only the insights it was able to impart to its human partner.

The way these seemingly disparate goals can be expressed as reflections of the same idea gives rise to the central thesis of this dissertation:

Thesis 1. *The primary goal of interpretable AI is to improve the robustness of human decision-making in the presence of an AI model.*

Another way of articulating this goal is that the purpose of interpretability is to improve the performance of the hybrid system that implicitly emerges whenever a human overseer interacts with an AI model (Figure 1.1). By allowing the overseer to gain a better understanding of the model's reasoning, we hope that interpretability allows them to make a higher-quality decision about when to trust or distrust the

model.

All of the work described in this document proceeds from the assumption that this robustness objective is the ultimate goal of interpretability, and I describe the contributions herein in terms of their progress toward this goal.

In particular, I focus in this dissertation on **local robustness**, which, mirroring the definition of local interpretability that I give below, means allowing humans to audit individual model decisions and, hopefully, catch classifier errors without accidentally tagging classifier successes as erroneous. This is distinct from a more global type of robustness that would apply to the decision to trust or distrust a model generally, or the choice of models between alternatives. A good example of the latter is Amazon's recent discovery (by examining feature importance weights) that a model trained to rate the quality of job applicants was biased against women (*Dastin*, 2018). However, I assume in this document the presence of a model which either static or as optimized as possible, and examine the potential for explanations to improve the utility of such a model.

1.1.4 Methods

The last several years have seen a huge proliferation of work on interpretability methods, particularly as applied to neural networks. There are several important conceptual divisions in the methods literature, including: global versus local explanations; explanation type; posthoc versus intrinsic interpretability; and input data type. I review these divisions as a means to provide a brief survey of the field.

1.1.4.1 Global versus Local Explanations

One of the most important distinctions in recent interpretability work is that between global and local interpretability. Global interpretability is generally considered to mean a model which can be understood by directly examining its parameters and

structure. Examples of models considered to be globally interpretable include linear models such as logistic regression and support vector machines, as well as simple tree-based methods such as decision trees (*Murdoch et al.*, 2019). In a logistic regression, for example, a human auditor can gain insight into the functioning of the model by looking at its top regression coefficients in order to understand which features have the greatest impact on the model when present. A similar inspection is possible with decision trees.

One hallmark of globally interpretable models is that they cannot exceed a certain level of complexity while still being subject to human examination. This limits their predictive power. So, while some recent work has focused on reducing complex models to simple, globally interpretable models (e.g. *Wu et al.* (2017)), it is generally believed that there is an unavoidable tradeoff between global interpretability and predictive power. As a result, most recent interpretability work has focused on local interpretability rather than global.

Local interpretability refers to models whose structure may be too complex to examine as a whole, but for which individual decisions can still be understood (*Guidotti et al.* (2018)). The most common example of local interpretability is the idea of local feature attribution. In this scenario, a complex model such as a deep neural net makes a decision on some input $\mathbf{x}^i$, and then some procedure is followed to determine which features x^i_j of the input had what impact on the output. What makes such an explanation "local" is that the relationships that are described, such as that feature x^i_j has a high positive impact on the output $\hat{y}^i$, may hold for that particular input but not for others. There might exist some other input $\mathbf{x}^n$ for which that same feature actually has a negative impact or no impact, because of its interactions with other features of the input. Furthermore, it might be impossible to succinctly describe the range of ways that feature x_j can affect the model output across the full input space.

1.1.4.2 Explanation Types: Feature-based and Example-based

Local explanations for model decisions fall broadly into two categories: feature-based and example-based. Feature-based explanations seek to clarify the relationship between the features x_j^i of an input $\mathbf{x}^i$ and the model output $\hat{y}^i$, while example-based explanations attempt to draw in information from outside that particular example-prediction pair, sometimes in the form of other input examples $\mathbf{x}^n$ which in some way contextualize the model's prediction on input $\mathbf{x}^i$. This basic distinction mirrors that which exists in statistical learning theory between model-based prediction methods like linear regression and model-less methods like k-nearest-neighbors (*Hastie et al.*, 2001b).

Within the ambit of feature-based explanations, feature attribution is by far the most common approach, sometimes described as saliency maps or rationales (*Guidotti et al.*, 2018). As discussed above, this type of explanation seeks to generate a parsimonious set of weights on the features of the input which describe how those features impacted the output of the model. One distinction between different approaches to feature attribution is what exact semantic meaning is attached to the weights that are generated by the method: if a feature x_j is assigned a high-magnitude weight z_j with respect to an output $\hat{y}$ for an input $\mathbf{x}$, there are a variety of things that weight can mean. It can mean that x_j had a high positive impact on $\hat{y}^j$ (e.g. (*Arras et al.*, 2017), or that it simply had a high impact on the output, regardless of sign (e.g. (*Simonyan et al.*, 2013)). This latter interpretation is usually what is referred to as saliency.

Another distinction between feature attribution methods is how they deal with feature collinearity. (*Li et al.*, 2016) generates each attribution mask $\mathbf{z}$ by finding the minimum subset of features which, when erased from the input, causes the model's output class to flip. (*Lei et al.*, 2016), by contrast, finds the minimum set of features necessary for the model to produce an output $\hat{y}$ which is similar to the target value

y. While this is a subtle difference, it can lead to different attribution weights with different interpretations. While there has been some high-level discussion of feature attribution as a general approach (e.g. *Ancona et al.* (2018); *Galassi et al.* (2019)), there have been so many papers on this topic and so little high-level synthesis that there does not yet exist a comprehensive taxonomy of feature attribution methods which enumerates their relative strengths, weaknesses, and semantic differences.

While feature attribution methods dominate the literature on methods for feature-based interpretable machine learning, there are other explanation types that have been explored. Rule-based explanations seek to identify logical rules that hold reasonably reliably over the operation of complex models at either a local or a global level (e.g. *Ribeiro et al.* (2018); *Lakkaraju et al.* (2017)). Natural language explanations seek to articulate model reasoning in the form of textual explanations (*Ehsan et al.*, 2018).

Example-based explanations are much less well-represented than feature-based explanations in the contemporary interpretable ML literature. They generally seek to draw in information from beyond the very narrow scope of the given input example x and the model's prediction $\hat{y}$ in order to provide evidence for, against, or which otherwise helpfully contextualizes that prediction.

The classical example-based *prediction* (as distinct from interpretability) algorithm is k-nearest-neighbors, which makes a prediction about an input $\mathbf{x}^j$ by reference to the target values of $\mathbf{x}^j$'s neighbors within the input space $\{\mathbf{x}^n\}$. Also of note is the venerable field of case-based-reasoning (CBR) which, broadly construed, "addresses new problems by remembering and adapting solutions previously used to solve similar problems" (*Goel and Diaz-Agudo*, 2017). CBR is a broad literature which predates contemporary machine learning and remains independent of the general thread of machine learning research, though it does frequently overlap with it. However, much of CBR research pertains to the making of correct decisions based on existing evidence

(cases), and not to the explanation of existing algorithms per se. There has been some recent discussion in the CBR literature of the "twinning" of CBR ideas with machine learning models (*Keane and Kenny*, 2019).

Within the recent literature on interpretable ML, a somewhat representative example-based algorithm is (*Koh and Liang*, 2017), which uses the Hessian matrix of a model's loss function to identify points that were highly influential in training that model to make its decision on input x^j. Related but slightly different to example-based algorithms are prototype-based algorithms, which seek to reduce the dataset to a small set of "prototype" examples which are thought to represent the entire dataset and which can then be used as justification for model decisions (*Kim et al.*, 2016; *Li et al.*, 2017).

Finally, there is a small line of literature on what could be called "concept-based" explanations, which perform clustering of input features into human-understandable "concepts", and then try to explain classifier decisions in terms of these concepts. In a sense, this style of approach is a hybrid between feature- and example-based approaches, because it synthesizes the dataset into useful feature clusters and then uses these clusters to explain model decision. Examples of this approach include (*Ghorbani et al.*, 2019; *Kim et al.*, 2017; *Chen et al.*, 2018).

1.1.4.3 Posthoc versus Intrinsic Interpretability

A very important distinction in the contemporary interpretable machine learning literature is that between posthoc and intrinsic (or model-based) interpretability, sometimes framed as "explain" versus "interpret" (*Rudin*, 2019). Posthoc explanations seek to explain an existing model, while intrinsic model-based explanations seek to engineeer models that incorporate interpretability into their reasoning in ways which can be directly audited (*Murdoch et al.*, 2019).

Posthoc explanations can be further subdivided into analytic and perturbation-

based methods. Analytic methods, sometimes also referred to as gradient-based methods, are exclusive to neural nets. They typically involve some variant of the idea of mathematically decomposing the output of the model in order to understand how each input contributed to said output (*Ancona et al.*, 2018).

Perturbation methods, by contrast, involve directly calculating feature impacts by perturbing the input and seeing how the output changes. These include the very popular LIME method (*Ribeiro et al.*, 2016), which constructs a local explanation for a model by training a linear approximation to that model on perturbations around the input-of-interest. SHAP (*Lundberg and Lee*, 2017) is another example, claiming to represent a generalization of existing methods including LIME, deepLIFT (*Shrikumar et al.*, 2016) and layerwise relevance propagation (LRP) (*Bach et al.*, 2015).

Posthoc methods have the advantage of being applicable to already-trained models, meaning that engineers can apply them to their existing models rather than having to train intrinsically interpretable models that may suffer in performance compared to conventional ones. Perturbation methods in particular are often model agnostic, working as well for a random forest as for a convolutional or recurrent neural network.

However, both types of posthoc explanation method have drawbacks. Analytic methods such as LRP (*Bach et al.*, 2015) are often specific to a particular neural architecture and difficult to adapt to novel architectures, while perturbation methods can be prohibitively slow since they often involve running an iterative procedure on every point that needs to be explained. Furthermore, explanation procedures that retroactively analyze the reasoning of a non-interpretable model can sometimes reveal pathological behavior that leads to incomprehensible explanations (*Feng et al.*, 2018).

Model-based interpretability, by contrast, seeks to engineer models which have intrinsic interpretability. A common approach to this goal is the idea of model attention, in which models selectively choose to focus more on certain portions of a given input than others, generally with a sparsity constraint that encourages them

to attend to as little as possible of a given input. These attention weights can then be examined in much the same way as those produced by a posthoc method (*Galassi et al.*, 2019; *Guidotti et al.*, 2018). In addition to improving interpretability, neural attention has been found to improve model performance on certain tasks such as machine translation (*Luong et al.*, 2015), and attention-based models are a crucial component of the current state-of-the-art on a number of NLP tasks (*Devlin et al.*, 2018).

Model attention has advantages and disadvantages in comparison with posthoc methods. One advantage is that attention scales well in comparison to perturbation methods because attention weights are generated as part of the model's ordinary functioning rather than as a result of additional processing and analysis. Another advantage of attention is that it can be manipulated via the model objective function to actually force models to reason in a more interpretable way without necessarily reducing their performance (*Rudin*, 2019). This can potentially represent a solution to the types of pathological behavior observed in *Feng et al.* (2018).

However, model attention represents an estimate by the attention layer of feature importance. Just because the attention layer chooses to attend to a feature does not necessarily mean that that feature is impactful in driving the model's prediction. This observation and others have lead to an ongoing debate in the interpretability literature about the legitimacy of model attention as an explanatory mechanism (*Jain and Wallace*, 2019; *Serrano and Smith*, 2019; *Vashishth et al.*, 2019; *Wiegreffe and Pinter*, 2019).

1.1.4.4 Input Data Type

A final important conceptual division in contemporary interpretability literature is that of input data type. Interpretability work has tended to be pioneered on image data and then transferred to other domains such as text and tabular data. Other data

types, such as time series data, have seen relatively little attention in the literature (*Guidotti et al.*, 2018).

Image data is composed of pixels which are individually meaningless but which, when clustered appropriately, form themselves into visual concepts that are easily recognizable by humans (e.g. "wing", "dog", "chair"). Text data, by contrast, is generally modeled as being composed of tokens which have meaning, but which don't group naturally into easily recognizable concepts. For this reason, works such as (*Chen et al.*, 2018) which decompose images into visual concepts and then explain model prediction in terms of those concepts are appropriate for image data, but less so for text data.

Another difference between these two genres of data is that image recognition tasks generally involve fewer causal pathways than textual data. That is, the most basic reason why an image would be a picture of a dog is because it has a dog in it somewhere. Text data, by contrast, can arrive at a typical text classification category via a number of routes: a sentiment recognition model might deem a text to be "angry" for any number of reasons, from the use of angry words, to the use of excessive exclamation points, to the use of all-caps typing. Two equally "angry" texts can share few or no tokens in common.

As a result of this contrast, works such as (*Kim et al.*, 2016) which find a small set of "prototype" examples to represent each class, are more appropriate for image data than for text data–it makes sense to explain a prediction of "dalmation" by showing a single prototypical dalmation image, but it would be difficult to generate or identify a single prototypical example of "angry" that would serve as a suitable explanation for any instance of anger in a sentiment detection task.

Tabular data has different properties altogether from text and image data. As discussed above, image- and text-classification tasks often have different conceptual hierarchies that make certain kinds of explanation methods more or less appropri-

ate: tabular data often lacks any kind of recognizable conceptual hierarchy at all. (*Poursabzi-Sangdeh et al.*, 2018) investigates the effect of interpretable machine learning on the ability of humans to make accurate predictions about the prices of apartments. Their results are largely negative, and they conclude that humans simply don't have strong enough intuitions about how apartment qualities factor into apartment pricing to be able to benefit from simpler, better explained models.

Tabular data is also more likely to be involved in prediction tasks rather than classification tasks per se (*Hastie et al.*, 2001a). What that means is the modeling of tabular data often contains a degree of intrinsic randomness in the outcome that is lacking in tasks such as image and text classification. Hence, interpretability techniques for tabular data need additionally to express this underlying uncertainty, a topic which has seen some discussion in the HCI literature (e.g. (*Kay et al.*, 2015)), but has yet to make its way into the main stream of contemporary interpretability work.

1.1.4.5 Summary

The conceptual divisions listed above are broadly orthogonal with one another. For example, prototype-based explanations can be seen as a global form of example-based explanations in that they involve defining a limited set of representative prototypes for each class across the entire input space. Both feature-based and example-based explanations can be generated in either post-hoc or intrinsic manners, although only intrinsically feature-based prediction algorithms can produce intrinsic feature-based explanations and vice versa.

The division of input data type stands out from the other three listed divisions in that it is more a consideration than a categorization per se–any type of explanation can be generated for any input type, but certain types of explanations are more naturally suited to certain input data types.

1.1.5 Evaluation

Doshi-Velez and Kim (2017) group evaluations of interpretable machine learning into three categories:

1. **Functionally-grounded**: No humans are involved; evaluation based on proxy metrics (e.g. measuring explanation sparsity).
2. **Human-grounded**: Human subjects attempt a simplified or indirect task using interpretable machine learning (e.g. user satisfaction surveys).
3. **Application-grounded**: Human subjects attempt a real decision task using interpretable machine learning (e.g. diagnosing model errors).

Every new interpretability algorithm is presented with some form of evaluation, usually falling into one or the other of the first two categories. The most common evaluation for a new technique is to define a proxy quality such as "fidelity" or "interpretability" (read sparsity) and then to perform an automated evaluation of the proposed algorithm on these qualities in comparison to competing algorithms (e.g. *Lakkaraju et al.* (2017); *Arras et al.* (2017); *Shrikumar et al.* (2017)). This type of evaluation is often presented in conjunction with one or more case studies showing anecdotal examples of the proposed algorithm's beneficial qualities (e.g *Kim et al.* (2016); *Li et al.* (2017); *Chen et al.* (2018)).

Less commonly will a methods paper include a user study. These studies sometimes directly measure subject performance outcomes like decision quality, but often measure human performance on proxy tasks such as simulation or indirect metrics such as user satisfaction. These studies tend to be quite small scale ($n < 50$) (e.g. *Ribeiro et al.* (2016); *Kim et al.* (2016); *Lakkaraju et al.* (2016)).

A recent trend within the interpretability literature has been the emergence of work centered solely around user studies with the goal of teasing out the human factors of interpretability rather than pioneering new algorithms. These papers typically fall

into the "application-grounded" category of *Doshi-Velez and Kim* (2017), measuring human speed, accuracy and other outcomes on applied tasks with and without the benefit of explanations. These works have assessed the impact of interpretability on how humans detect deceptive online reviews(*Lai and Tan*, 2019), estimate apartment prices (*Poursabzi-Sangdeh et al.*, 2018), perceive the competence of a visual reasoning system (*Cai et al.*, 2019b), as well as their performance on synthetic decision tasks (*Lage et al.*, 2018; *Friedler et al.*, 2019). While this style of literature is growing, it still represents a minority of interpretability work, and there have been several calls for more rigorous human experimentation within this field (*Doshi-Velez and Kim*, 2017; *Abdul et al.*, 2018).

Because of the relative paucity of this type of work, there is no consensus yet on what constitutes an ideal experimental design for this type of study. The basic structure generally takes the form of a between-subjects experiment in which subjects are asked to interact with a model's predictions about individual items in some prediction task. Different experimental groups of subjects are generally exposed to different models or variants of the same model with different putative levels of interpretability, and their performance is measured on some outcome measure. This measure is often accuracy on the decision task at hand relative to a known ground truth, but sometimes other tasks such as ability to successfully simulate the outcome of the model, as discussed above.

Perhaps the biggest distinguishing feature between these types of work is their handling of the relative balance between human and model skill. Some studies (e.g. *Lage et al.* (2018); *Friedler et al.* (2019) focus on artificial decision tasks for which human subjects cannot by definition have existing intuitions. Others involve real-world prediction tasks for which humans have more (e.g. the sentiment detection task used by *Nguyen* (2018)) or less strong intuitions (e.g. the apartment price prediction task used by *Poursabzi-Sangdeh et al.* (2018)). Among studies based on

real world tasks, some choose to measure baseline unassisted human performance on the prediction task and some do not, posing themselves as purely a relative comparison between different interpretability conditions.

I argue that this balance of human and model skill is an as-yet undiscussed but crucial factor in the design of studies intended to evaluate the interpretability of predictive models. For example, in cases where the baseline unassisted performance of human subjects is much lower than that of the model (e.g. *Lai and Tan* (2019)), a big improvement to human predictive performance can be gained simply by persuading subjects to more readily accept model decisions.

A notable commonality between these works is that those of them which make the appropriate comparisons have generally failed to find a significant positive impact of explanations on human accuracy (e.g. *Lai and Tan* (2019); *Lage et al.* (2018); *Poursabzi-Sangdeh et al.* (2018); *Friedler et al.* (2019); *Bussone et al.* (2015); *Weerts et al.* (2019)). That is, none of the studies cited above have found that the presence of explanations improves human accuracy on the given prediction task. This is an interesting result because it represents a sharp contrast to the general enthusiasm with which interpretability methods work has been greeted by the machine learning community at large. The user studies described in chapters III and V continue this trend.

What it suggests is that the task of productively aligning human and machine decision making is extremely challenging, and much more than simply a question of inventing interpretability methods that demonstrate good fidelity, sparsity, etc. It suggests that there are deep human factors involved in what makes an effective interpretability technique, factors which can only begin to be teased apart by further human experimentation.

Finally, another commonality in the recent literature on this subject is that it largely pertains to feature-based explanations. *Lai and Tan* (2019) has one example-

based experimental condition, while *Cai et al.* (2019b) and *Cai et al.* (2019a) study example-based explanations but measure satisfaction rather than performance-based outcomes. The other cited studies involve only feature-based methods. The consistent failure of feature-based explanations to improve outcomes across multiple domains seems to imply that this type of explanation, which clarifies the relationship between the input and the model outcome, may simply not provide human users with enough information to make more accurate decisions about the items of interest in a given scenario.

If this is true, then example-based explanations may be necessary in some domains to produce real gains in human performance. This style of explanation can add the additional information that is lacking from feature-based explanations, in the form of examples that can provide evidence for or against a model's prediction on a given item-of-interest. However, this additional information represents an exponential increase in the size of the design space surrounding how to actually present explanations to users.

Beyond interpretability per se, the work presented here is an example of AI-advised human decision making, which has been shown to be a difficult and delicate partnership to enable (*Bansal et al.*, 2019). Even more generally it falls into a genre of literature which might be termed human-AI interaction, which has shown recently that intelligent-yet-opaque algorithms tend to inspire both discomfort and inordinate trust (*Springer et al.*, 2017; *Warshaw et al.*, 2015). This discomfort, at least, can be partially alleviated by increasing transparency (*Eslami et al.*, 2018).

1.2 Toxicity Detection

Throughout this work I use the task of detecting abusive social media content as my primary application domain for interpretable machine learning. Abusive language goes by many names and sub-categories in the literature, including hate speech

(*Fortuna and Nunes*, 2018), aggression (*Kumar et al.*, 2018), toxicity (*Wulczyn et al.*, 2017), cyberbullying (*Hosseinmardi et al.*, 2015), harassment (*Golbeck et al.*, 2017) and incivility (*Anderson et al.*, 2016). *Waseem et al.* (2017b) proposes a typology of different types of abuse. Because I primarily use models trained on the toxicity dataset introduced by *Wulczyn et al.* (2017), I generally use the term "toxicity" for the target outcome for the decision tasks considered in this paper, while acknowledging that toxicity is one among a number of related dimensions in this domain.

Toxicity detection has two qualities that make it particularly apt as a domain for exploring the utility of interpretable machine learning: 1) it is a worthwhile task with significant real world implications; 2) it is a task with strong potential for improvement from interpretable machine learning.

Toxicity detection is a worthwhile task–abusive language on social media is a major societal problem. Even when it doesn't directly harm its objects, it still causes harm by limiting the productivity of conversations about controversial topics like politics–it's impossible for ideologically opposed people to find common ground without a basic level of conversational civility (*Anderson et al.*, 2014). On many platforms, human moderators work to filter posts that violate community standards for civil conversation, but this is expensive, labor-intensive and vulnerable to biases, mistakes and fatigue among those moderators.

The salience of the issue has attracted a good deal of recent attention from the computational community. Scholarly work has assessed the prevalence and impact of online abuse (*Lenhart et al.*, 2016; *Anderson et al.*, 2014; *Pew*, 2016; *Anderson et al.*, 2016), while a number of studies have sought to construct datasets for its study and modeling (*Wulczyn et al.*, 2017; *Abbott et al.*, 2016; *Kennedy et al.*, 2017; *Napoles et al.*, 2017; *Golbeck et al.*, 2017). Annual workshops for the study of the issue have been established at several NLP and machine learning conferences (*Waseem et al.*, 2017a; *Kumar et al.*, 2018; *for Computing Machinery (ACM)*, 2016). Many papers

have been published on the application of machine learning to the detection of abusive online content (e.g. *Nobata et al.* (2016); *Fortuna and Nunes* (2018); *Cheng et al.* (2015); *Pavlopoulos et al.* (2017); *Chancellor et al.* (2017).

However, it is considered unlikely that a completely automated approach can provide a good solution to the problem. The task is subjective and context-specific. Different communities, for instance, have different norms for acceptable content (*Chandrasekharan et al.*, 2018; *Fiesler et al.*, 2018). Different individuals have different perceptions about what constitutes abuse with respect to linguistic features like profanity (*Malmasi and Zampieri*, 2018) or context (*Blackwell et al.*, 2018a). It is very easy for labeler bias to propagate into trained models (*Binns et al.*, 2017), while *Olteanu et al.* (2017) points out that traditional metrics like accuracy may belie the actual human impact of model errors. Toxicity classifiers are also easy to fool with nonstandard language (*Hosseini et al.*, 2017).

What appears to be needed is a way to combine the efficiency of automated approaches with the flexibility of human oversight. Examples that have begun to be explored in the literature include hybrid systems which query humans about low-confidence items (*Link and Hellingrath*, 2016; *Pavlopoulos et al.*, 2017) and using interface design to encourage bystanders to intervene in cyberbullying incidents (*DiFranzo et al.*, 2018).

This need for hybrid approaches is where interpretable machine learning can potentially improve the status quo for detecting toxic content on social media. In combination with ideas like reserving human judgement for low-confidence examples, interpretable machine learning has the potential to allow moderators to make quicker, more fair, more consistent judgments about content.

A few works have specifically pursued the idea of interpretable ML for abuse detection: *Svec et al.* (2018) shows that an interpretable model can match human-generated annotations with high precision, while *Pavlopoulos et al.* (2017) proposes

using explanations to help humans make decisions about borderline instances. *Wang* (2018) analyzes pitfalls associated with using interpretable ML for abuse detection, showing that recurrent neural nets suffer from certain kinds of pathological behavior that prevent conventional interpretation techniques from working well, behaviors which are also noted and explored by *Feng et al.* (2018).

In summary, the problem of abusive language on social media is a good testbed for interpretable machine learning because it is a recurrent text classification problem with significant real-world impact, for which some level of human oversight will probably always be necessary, but which may be able to benefit from machine learning. The potential of interpretable machine learning to efficiently hybridize human and machine effort could be the key to enabling this type of collaboration in this domain.

1.2.1 Generalizability

Given the focus on this one domain, a reasonable question is whether methodological advances are likely to generalize to other application areas. Is a feature attribution technique which is proven to help human annotators make more accurate toxicity judgments liable to help as an aid for automated essay grading?

I argue that proxy metrics such as attribution mask recall (as I use in the Chapter II empirical evaluation) **are** likely to generalize beyond the task of toxicity. If an attribution technique can be shown to do a good job of identifying tokens constituting predictive signal in one domain, there seems no strong reason to think that result would not generalize to another domain, except insofar as the model might struggle to perform classification in that domain in the first place.

Human outcomes are likely to be more domain-specific, since the extent to which human subjects benefit from the highlighting of toxic content is likely to differ from how they benefit from the highlighting of indicators of bad grammar in a student essay. In section 1.1.5 I discuss the effect of human-model skill complementarity.

This factor alone would cause big differences in the subject-model relationship across the domain, with humans having strong intuitions in some domains (e.g. toxicity) and not others (e.g. deceptive review detection), and likewise for models.

The question of how human results are likely to generalize is complicated somewhat by the fact that the contemporary applied interpretability literature has yet to demonstrate a significant positive result. What this means is that not only do we not know how a positive outcome in one domain is likely to transfer to another domain, we do not even know how to achieve a positive outcome at all.

1.3 Contributions

In this dissertation I propose two novel interpretability algorithms for neural text classifiers. Both methods are for local interpretation, meaning that they both involve trying to explain a classifier's prediction on a single item of interest $f(\mathbf{x}^i) = \hat{y}^i$.

The first method combines the idea of adversarial training with that of neural attention to produce attention masks which capture all available predictive signal in a given input. The second method combines the idea of feature attribution with that of example-based explanations to retrieve explanatory examples based only on the features that actually impacted the model prediction.

Each of these methods represents a novel contribution in its own right. The adversarial attention mechanism I propose brings a level of semantic clarity to neural attention which has previously been lacking from this particular approach to feature attribution (e.g. *Jain and Wallace* (2019)). The reason this represents a contribution to feature attribution as a whole is that neural attention is easier and more intuitive to manipulate via the model's objective function than gradient-based methods such as *Ross et al.* (2017), and perturbation methods cannot be manipulated at all. If we want attention masks to be more sparse, more cohesive or more comprehensive, these qualities can all be encouraged by adding different terms to the overall objective

function.

This quality of being able to place constraints on the model's feature attribution allows us to not only understand how it is reasoning, but to change the way it reasons entirely. As I discuss in Chapter VI, this flexibility opens up new avenues for manipulating model behavior beyond passive attribution.

The second proposed algorithm uses model attention to reduce an item-of-interest down to only the features that were predictive of the target class, and then retrieves analogous examples from the training data in order to contextualize the model's prediction on that item. The algorithmic contribution of this method is in producing examples which are visibly relevant to the item-of-interest while still retaining useful qualities as indicators of potential model error.

I evaluate both methods with rigorous, relatively large scale user studies that test their utility in terms of human performance, as well as revealing insights into the human factors driving the practical effectiveness of interpretable machine learning.

The first study evaluates the utility of the adversarial attention mechanism. We ask subjects to predict the consensus toxicity of a series of comments drawn from the dataset used to train the model. Different subject groups perform this task with varying levels of algorithmic assistance, ranging from no assistance to both the prediction and attention mask produced by the adversarial attention model described above. I find that while these attention masks reduce the cognitive burden associated with the visible presence of a model prediction, they do not help subjects make better decisions about the correctness of those decisions (though they do change the distribution of human error).

Beyond its outcome, this study represents a contribution to the literature on designing evaluation studies for interpretable machine learning. I argue that several of the steps we take in this study, such as performing stratified sampling of comments for subjects to label, and recollecting the existing ground-truth value, address threats

to validity that would otherwise exist in this type of study.

The second study evaluates both the desirability and utility of the proposed attention-based explanatory example retrieval method. Subjects perform a similar task to that of the first study–predicting the consensus toxicity of social media comments. In one experimental group, subjects are asked to choose between examples produced by several retrieval algorithms: our proposed method and two baselines. We find that subjects prefer our method to the baselines by a wide margin.

In a second experiment with the same items, subjects are asked to use examples retrieved by a single algorithm to attempt to identify and correct classifier errors via analogous reasoning. We find that none of the algorithms we test, neither our proposed algorithm nor either baseline, are able to improve human predictive performance on this task. However, by overcoming the basic problem of identifying relevant text examples in the first place, we are able to identify several design issues that would otherwise be masked, such as the importance of both diversity and representativeness in selecting examples to display to a subject.

Taken together, the two proposed algorithms are a way to exploit the entire feature-example data matrix underlying a machine learning model (Figure 1.2). The adversarial attention mechanism provides a way of identifying a small set of important feature columns and clarifying their relationship with the model output. The example retrieval method allows for the identification of analogous example rows which can provide useful insights about the model's prediction on the item-of-interest.

As summarized in section 1.1.5, the literature, including our first user study, has generally found that feature-based explanations have no impact or only a marginal positive impact on human accuracy in troubleshooting classifier predictions. While our initial foray into the potential for example-based explanations for this purpose finds a similar negative result, I argue nevertheless that this type of method is a promising research direction for applied interpretability work.

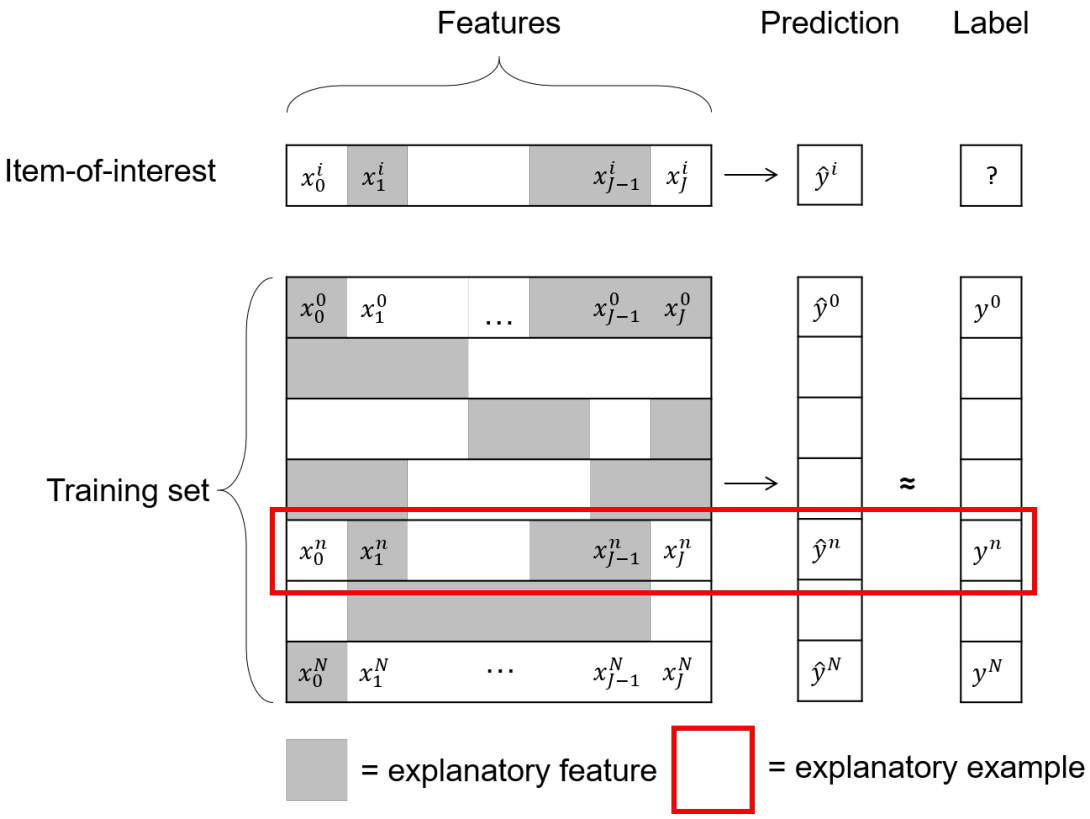

Figure 1.2: Diagram of information available to an explanatory system. Both features (grey) and examples (red) can be identified to explain a prediction.

CHAPTER II

Adversarial Attention for Feature Attribution

2.1 Introduction

As discussed in Chapter I, feature attribution is a common way to explain classifier decisions. In this style of explanation, sometimes known as rationales, saliency masks, or feature importance, a small subset of input features are identified as having been particularly impactful on the model output, in the form of a set of weights $\mathbf{z}$ of the same dimensionality of the input $\mathbf{x}$ (*Guidotti et al.*, 2018). This type of local explanation may not completely elucidate why a given example is assigned a given outcome, but it does simplify the relationship by identifying what attributes were considered in the decision

Feature attribution generally comes in three flavors: attention methods, analytic methods and perturbation- based methods. In attention methods, the model produces this attribution mask $\mathbf{z}$ as an additional output to its prediction $\hat{y}$. In analytic methods, the model is mathematically probed to estimate the impact of input feature, often utilizing the error or output gradients of the model. Finally, perturbation-based methods involve estimating feature importance by perturbing a given input

[0]This chapter consists of content published in
Carton, S., Mei, Q. and Resnick, P. (2018). Extractive Adversarial Networks: High-Recall Explanations for Identifying Personal Attacks in Social Media Posts, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.*

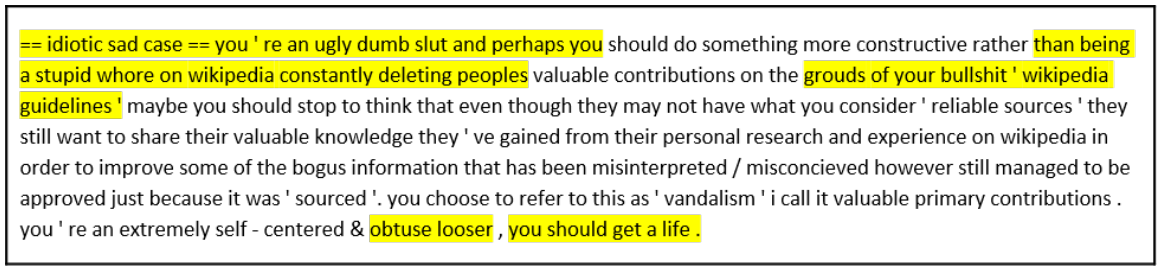

Figure 2.1: An example of a highly-attacking comment from the test set, rationalized by the model

and observing resultant changes in model output.

The advantage of attention methods among these approaches is that they can be controlled via the model objective function. If we want the attention mask to be more sparse, more cohesive, etc., these properties can be controlled by adding weighted terms to the model objective function controlling this aspect of model output. There has been work that does something similar for model error gradients (*Ross et al.*, 2017).

However, attention methods have certain drawbacks when used as feature attribution methods. *Jain and Wallace* (2019), for example, report poor alignment between model attention and other attribution methods. One possible explanation for this gap is that existing work on this topic has not explicitly addressed the problem of local feature redundancy. That is, when two features are equally predictive of an outcome, which of them should be included in the attribution mask for that decision? Typical sparsity constraints encourage minimal sufficient masks–unveiling just enough of the

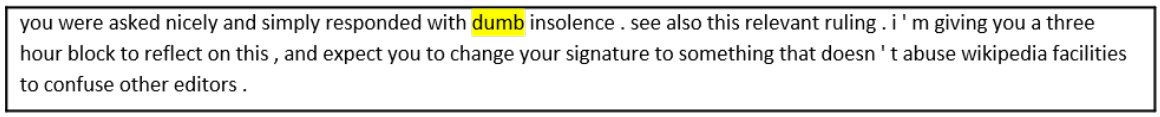

Figure 2.2: An example of a not-very-attacking example from the test set, rationalized by the model

example to justify the outcome. This can lead to ambiguously-defined or incomplete masks that may not accord well with the true semantics of the decision task.

As a solution to this ambiguity, we propose an adversarial mechanism for neural attention which is optimized to identify all potential predictive signal in a given input. We apply this model to the task of identifying which social media comments contain personal attacks and which words in those comments are the basis for classifying them as containing personal attacks. We train this model on a large dataset (*Wulczyn et al.*, 2017) of comments labeled for the presence of such attacks, and use the explanatory capacity of the model to identify spans that constitute personal attacks within those comments. We extend the work of (*Lei et al.*, 2016) in using one recurrent neural net (RNN) to produce an explanatory hard-attention rationale and a second RNN to make a prediction, the two models trained in an end-to-end fashion.

The adversarial mechanism works as follows: to produce complete (i.e. high-recall) explanations, we add to this existing architecture a second, adversarial predictive layer whose purpose is to try to make predictions based on what is left out of the rationale. We then add a term to the attention layer objective function which encourages it to fool this secondary predictive layer into making poor predictions by including all predictive signal (i.e. personal attacks) in the mask that it generates.

We also show that manipulating the model bias term to set a semantically appropriate "default behavior" or "null hypothesis" for the model significantly improves performance. That is, by explicitly choosing what output a zero-information, empty explanation should correspond to, the model is able to learn explanations that correspond more closely with human-generated data.

In an empirical evaluation, we collect a dataset of human judgments about which spans of texts constitute personal attacks in a subset of the *Wulczyn et al.* (2017) dataset. We show that our proposed algorithm achieves both better precision and better recall at the token level than existing baselines, demonstrating the effectiveness

of the adversarial approach to feature attribution.

To summarize, the contributions of this chapter are as follows:

- We articulate feature attribution as an adversarial problem and introduce an adversarial scheme for extraction of complete (high-recall) attribution masks for text classifier decisions.
- We demonstrate the value of explicitly setting a default output value in such an explanatory model via bias term manipulation.
- We apply explanatory machine learning for the first time to the task of detecting personal attacks in social media comments, and develop a validation dataset for this purpose.

2.2 Model

Given the application domain of detecting personal attacks, the goal of the proposed architecture is to highlight personal attacks in text when such are present, and to highlight little or nothing when there are none, while also performing accurate overall prediction.

These requirements prompt two important edge cases. First, there may be no particular predictive signal in the comment text (i.e. no personal attacks); in a more typical explanatory setting there is always assumed to be some explanation for a decision. Second, there may be redundant signal (i.e. multiple personal attacks), more than is strictly required for accurate prediction, and we assume that it is desirable to identify all of it. We address both of these cases with modifications to the original model architecture.

The model (Figure 2.3A) is a hard attention architecture which uses one RNN to extract an attention mask of either 0 or 1 for each token, and a different RNN to make a prediction from the attention-masked text (detailed in Figure 2.3B). Following (*Lei*

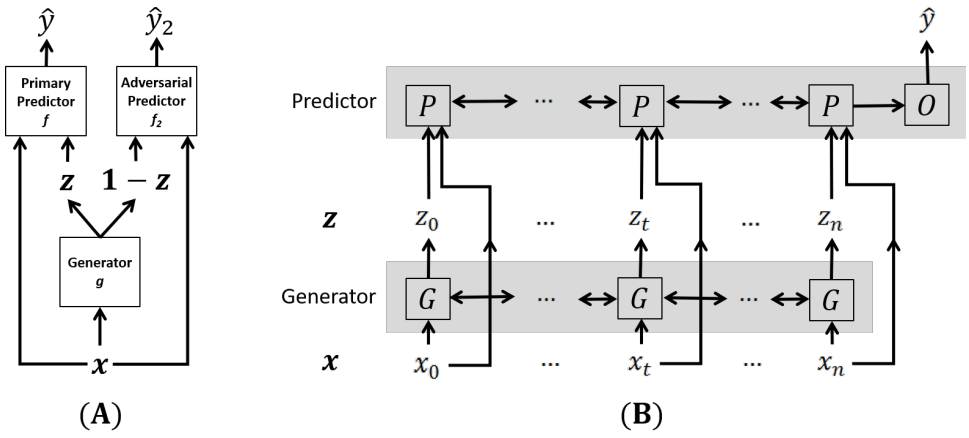

Figure 2.3: (A) Overall architecture. Generator and predictors are RNNs; (B) Detail of interaction between generator and one predictor layer. G and P are recurrent units of any kind. O is a sigmoid output layer.

et al., 2016), we refer to the mask-producing layer g as the *generator*, but for clarity we call the predictive layer f the *predictor* rather than the encoder. Again following previous work, we refer to the output $\mathbf{z}$ of the generator as the *rationale*, in that it rationalizes the prediction of the predictor. We also refer to the inverse rationale, defined as $1 - \mathbf{z}$, as the *antirationale*.

To this basic two-layer scheme, we add a secondary, adversarial predictor f_2, which views the text masked by the antirationale rather than the rationale. The secondary predictor's role is to act as an adversarial discriminator–it tries to make accurate predictions on the antirationale, while the generator tries to prevent it from doing so, which ensures that all predictive signal ends up in the rationale.

2.2.1 Primary predictor

The primary predictor f is an RNN which views the input text masked by the rationale produced by the generator. Its objective is simply to reduce its own squared loss:

$$cost_f(\mathbf{z}, \mathbf{x}, y) = \left[f(\mathbf{x}, \mathbf{z}) - y\right]^2 \tag{2.1}$$

2.2.1.1 Default behavior via predictor bias term manipulation

The default behavior of the model is the prediction the predictor makes if the input is entirely masked by the rationale: $f(\mathbf{x}, 0)$. When working with a recurrent unit that has no internal bias term, this behavior is entirely determined by the bias term of the final sigmoid output layer, $\sigma(\mathbf{wx} + \mathbf{b})$, which with typical random initialization of b results in a default predicted value of roughly 0.5.

However, this 0.5 default value is not always optimal or semantically appropriate to the predictive task. In the personal attack detection task, if no attacks can be detected, the "natural" default target value for a text should be close to 0. We show in the experiments that manually setting the output layer bias term b to $logit(0.05) = -2.94$, so that the default predicted value is 0.05, improves model performance.

2.2.2 Secondary adversarial predictor

The secondary adversarial predictor is an RNN which views the input text masked by the antirationale, defined as 1 minus the rationale z. Its purpose is to encourage high-recall explanations by trying to make accurate predictions from the antirationale, while the generator tries to prevent it from doing so.

(A)

Comment with antirationale									y
moron	,	please	leave	me	alone	.	jerk	.	1
	you	are	a	real	piece	of	work	.	1
		why	did	you	revert	my	edits	?	0
this	article	is	a	good	piece	of	work	.	0
don	'	t	jerk	me	around	,	thanks	.	0

(B)

Comment with permuted antirationale									y	Permute?
moron	,	please	leave	me	alone	.	jerk	.	1	No
	you	are	a	real	piece	of	work	.	1	Yes
		why	did	you	revert	my	edits	?	0	No
this	article	is	a	good	piece	of	work	.	0	No
don	'	t	jerk	me	around	,	thanks	.	0	Yes

Figure 2.4: (A) Fabricated sample batch masked by antirationales. Note the correlation between mask and target; (B) The batch with some antirationales switched with those of other items. The correlation no longer holds.

However, if the adversarial predictor's objective function were simply $\left[f_2(\mathbf{x}, 1 - \mathbf{z}) - y\right]^2$, it would be able to gain an unfair advantage from the presence of masking in the antirationale. Seeing evidence of "blanked-out" tokens would tell it that personal attacks were present in that comment, giving it strong hint that the target value is close to 1.0 and vice-versa (see figure 2.4A).

To take away this advantage, the input to the adversarial predictor has to be permuted such that the mask itself is no longer correlated with the target value, while still allowing it to scan the antirationale for residual predictive signal.

Our solution is to replace the masks of half the items in a training batch with the masks of other items in the batch. We order the batch by target value. If item $\mathbf{x}^i$ is selected for replacement, it gets the mask of item $\mathbf{x}^{N-i}$ where N is the size of the batch. We call this permutation function c:

$$c(\mathbf{z}^i) = c(g(\mathbf{x}^i)) = \begin{cases} g(\mathbf{x}^i) & \text{if } k^i = 1 \\ g(\mathbf{x}^{N-i}) & \text{if } k^i = 0 \end{cases}$$

$$\mathbf{x}^i \in \{\mathbf{x}^0, ..., \mathbf{x}^N\} \qquad k^i \sim Bernoulli(0.5)$$

This ensures that low-target-value items get masks associated with high target values and vice-versa, to maximize the dissociation between masks and target values. Figure 2.4B demonstrates an example of such permutation. This may slow down the learning, since the adversarial predictor will sometimes have access to somewhat different features of the input than it will have on the test data, but it should not lead to incorrect learning, since the training data always has the correct label, regardless of the mask.

With $c(1 - \mathbf{z})$ as the permuted antirationale resulting from applying this randomization process. The objective for the secondary, adversarial predictor is its predictive

accuracy on this permuted antirationale:

$$cost_{f_2}(\mathbf{z}, \mathbf{x}, y) = \big[f_2(\mathbf{x}, c(1-\mathbf{z})) - y\big]^2 \tag{2.2}$$

2.2.3 Generator

Given that the two predictors are trying to minimize error on the rationale and (permuted) antirationale respectively, the objective function for the generator is as follows:

$$\begin{aligned} cost_g(\mathbf{z}, \mathbf{x}, y) = & \qquad (3) \\ & \big[f(\mathbf{x}, \mathbf{z}) - y\big]^2 \qquad (3.1) \\ & +\lambda_1 ||\mathbf{z}|| \qquad (3.2) \\ & +\lambda_1\lambda_2 \sum_t |z_t - z_{t-1}| \qquad (3.3) \\ & +\lambda_3 \big[f_2(\mathbf{x}, 1-\mathbf{z}) - f_2(\mathbf{x}, 0)\big]^2 \qquad (3.4) \end{aligned}$$

Terms 3.1-3.3 are present in the model of *Lei et al.* (2016). Term 3.1 encourages the generator to allow the primary predictor to make accurate predictions, prevents it from obscuring any tokens that would prevent the predictor from doing so. Term 3.2 encourages the generator to produce minimal rationales; obscuring as many tokens as possible. Term 3.3 encourages rationale coherence by punishing the number of transitions in the rationale; it encourages few contiguous phrases rather than many fragments in the rationale.

In theory, these three terms ensure high precision, selecting the minimal (term 3.2) rationale with sufficient signal for accurate prediction (term 3.1), subject to a coherence constraint (term 3.3).

Term 3.4, which is new, ensures recall by encouraging the adversarial predictor's prediction on the antirationale to be similar to the prediction it would make with

no information at all (aka the default value). That is, **the antirationale should contain no predictive signal**. Any personal attacks left out of the rationale would appear in the antirationale, letting the adversarial predictor make a more accurate prediction, which would be penalized by term 3.4.

2.2.4 Extractive Adversarial Network

In the GAN framework (*Goodfellow et al.*, 2014), a discriminator attempts to accurately classify synthetic examples which a generator is striving to match to the distribution of the true data. In our framework, the adversarial predictor attempts to accurately classify censored examples which the generator is striving to strip of all predictive signal. The discriminator in the GAN framework is trained half on real data, and half on fakes; our adversarial predictor is trained half on correctly-masked items and half on items with permuted masks. Where our framework differs from GAN is instead of generating adversarial examples which are compared to true examples, our architecture extracts a modified example out of an existing example, and so can therefore be described as an Extractive Adversarial Network (EAN).

2.2.5 Implementation details

For comparability with the original algorithm, we use the same recurrent unit (RCNN) and REINFORCE-style policy gradient optimization process (*Williams*, 1992) as *Lei et al.* (2016) to force the generator outputs to be a discrete 0 or 1. In this framework, the continuous output of the generator on each token is treated as a probability from which the mask is then sampled to produce a discrete value for each token. The gradient across this discontinuity is approximated as:

$$\frac{\partial \mathrm{E}_{z\sim g(x)}[cost_g(\mathbf{z},\mathbf{x},y)]}{\partial \theta_g} = \mathrm{E}_{z\sim g(x)}\left[cost_g(\mathbf{z},\mathbf{x},y)\frac{\partial \log p(\mathbf{z}|\mathbf{x})}{\partial \theta_g}\right]$$

In theory, one would sample z several times from the generator g to produce a good estimate of the gradient. In practice, we find that a single sample per epoch is sufficient. The predictors f and f_2 are trained as normal, as the error gradient with respect to their parameters is smooth.

We employ a particular hard attention model, but the idea of an adversarial critic is not limited to either hard attention or any particular recurrent unit. In a soft attention setting, our adversarial scheme will actually encourage "harder" attention by encouraging any non-zero attention weight to go to 1.0 (or else the inverse of that weight will leave predictive signal in the anti-explanation).

The attention weights produced by the generator are applied to the predictor at the output rather than the input level. When the recurrent unit P of the predictor operates on a token x_t modified by attention weight z_t, it ingests x_t normally, but depending on z_t it either produces its own output or forwards that of the previous token:

$$P(x_t, z_t) = z_t P_{base}(x_t) \cdot (1 - z_t) P_{base}(x_{t-1})$$

We investigate a similar range of sparsity hyperparameter values as the original model [1]. The weight on the inverse term only matters relative to the model sparsity, as that term cooperates rather than competing with the predictive accuracy term (because it almost never hurts accuracy to add more to the rationale). Therefore we set λ_3 to 1.0 when we want to include the inverse term.

We use Word2Vec (*Mikolov et al.*, 2013) to create input token word vectors and Adam (*Kingma and Ba*, 2014) for optimization.

[1] λ_1=[0.0003, 0.0006, 0.0009, 0.0012, 0.0015, 0.0018, 0.0021], λ_2=[0, 1, 2]

2.3 Data

To train our model of personal attacks, we use the dataset introduced by (*Wulczyn et al.*, 2017), which consists of roughly 100,000 Wikipedia revision comments labeled via crowdsourcing for aggression, toxicity and the presence of personal attacks. This dataset includes its own training, development and test set split, which we also use.

To this dataset we add a small validation set of personal attack rationales. 40 undergraduate students used Brat (*Stenetorp et al.*, 2012) to highlight sections of comments that they considered to constitute personal attacks. Comments were sampled in a stratified manny by selecting even numbers from each toxicity decile, from the development and test sets of the Wulczyn et al. dataset, and each student annotated roughly 150 comments, with each comment viewed by roughly 4 annotators. To calculate gold-standard rationales, we take the majority vote among annotators for each token in each comment. 1089 distinct comments were annotated, split between a development and test set of 549 and 540 examples respectively.

The Krippendorff's alpha on our validation set is 0.53 at the whole-comment level, meaning that annotators agreed at this level on whether they found any personal attacks at all in a given comment. This value is comparable with that of *Wulczyn et al.* (2017) (0.45). Agreement at the token level is a lower 0.41, because this includes tokens which are a matter of preference among annotators, such as articles and adverbs, as well as content tokens.

2.4 Empirical Evaluation

We conduct an empirical evaluation of the proposed algorithm in terms of its ability to match human effort in identifying which tokens in a comment are parts of personal attacks. This would not necessarily be appropriate in all domains, but in this particular task of predicting whether a comment has any personal attacks

in it, we argue that it is reasonable to define a good attention mask as one which contains all the tokens that were elements of personal attacks. We use the validation set described in the previous section to evaluate the performance of our algorithm on this objective.

We show that both modifications to the original algorithm, bias term manipulation and adversarial predictor, increase the tokenwise F1 of the predicted rationales relative to our human-annotated test set. All hyperparameters were tuned to maximize tokenwise F1 on the development set.[2]

2.4.1 Baselines

We generate six baselines for comparison with our variant of the (*Lei et al.*, 2016) architecture. These include the following:

Sigmoid predictor (logistic regression): Bag-of-words representation with a sigmoid output layer.

RNN predictor: The same sequence model used for the predictor, but with no generator layer.

Mean human performance: The mean tokenwise performance of human annotators measured against the majority vote for the comments they annotated (with their vote left out).

Sigmoid predictor + feature importance: Bag-of-words representation with sigmoid output layer, with post-hoc feature importance based on model coefficients. Cutoff threshold for features tuned to maximize rationale F1 on development set.

RNN predictor + sigmoid generator: Rationale mask generated by sigmoid layer applied independently to each input token. Prediction layer is same as predictor.

RNN predictor + LIME: Rationale mask generated by applying LIME (*Ribeiro et al.*, 2016) post-hoc to RNN layer predictions. Masking threshold tuned to maximize

[2] λ_1=0.0006 for variants without inverse term, λ_1=0.0015 for variant with inverse term, λ_2=2 (Tuned for maximum F1 on original model, then held constant for comparability)

rationale F1.

2.4.2 Rationale performance

In the main experiment, we evaluate model rationales relative to rationales created by human annotators. In our validation dataset, human annotators typically chose to annotate personal attacks at the phrase level; hence in the sentence "Get a job, you hippie s***bag", the majority-vote rationale consisted in our validation set of the entire sentence, where it could arguably consist of the last two or even the last word. Therefore, in addition to tokenwise precision, recall and positive F1, we also report a relaxed "phrasewise" version of these metrics where any time we capture part of a contiguous rationale chunk, that is considered a true positive.

We report results for the original model (i.e. terms 3.1-3.3 in the objective function), the original model with its bias term set for a default value of 0.05, and the bias-modified model with the additional inverse term (term 3.4). For every model variant, we optimized hyperparameters for tokenwise F1 on the development set. We also report results for the baselines described above.

Model	**Rationale**						**Prediction**		
	Tokenwise			**Phrasewise**					
	F1	**Pr.**	**Rec.**	**F1**	**Pr.**	**Rec.**	**MSE**	**Acc.**	**F1**
Sigmoid predictor	-	-	-	-	-	-	0.029	0.94	0.74
RNN predictor	-	-	-	-	-	-	0.018	0.95	0.78
Mean human performance	0.55	0.62	0.57	0.72	0.78	0.69	-	-	-
Sigmoid predictor + feature importance	0.20	**0.62**	0.12	**0.64**	**0.59**	0.70	0.029	0.94	0.74
RNN predictor + sigmoid generator	0.29	0.22	0.45	0.31	0.19	0.92	0.038	0.91	0.70
RNN predictor + LIME	0.33	0.29	0.39	0.4	0.25	**0.96**	0.018	0.95	0.78
Lei2016	0.44	0.38	0.52	0.51	0.38	0.83	0.021	0.95	0.77
Lei2016 + bias	0.49	0.48	0.49	0.60	0.46	0.86	0.02	0.95	0.77
Lei2016 + bias + inverse (EAN)	**0.53**	0.48	**0.58**	0.61	0.47	0.87	0.021	0.95	0.77

Table 2.1: Rationale performance relative to human annotations. Prediction accuracy is based on a binary threshold of 0.5. Performance of both Lei2016 model variants is significantly different from the baseline model (McNemar's test, $p < 0.05$)

Table 2.1 displays the results. The difference in performance between the three

baselines that don't use a RNN generator and the three model variants that do demonstrates the importance of context in recognizing personal attacks within text. The relative performance of the three variants of the Lei et al. model show that both modifications, setting the bias term and the addition of the adversarial predictor, lead to marginal improvements in tokenwise F1. The best-performing model approaches average human performance on this metric.

The phrasewise metric is relaxed. It allows a contiguous personal attack sequence to be considered captured if even a single token from the sequence is captured. The results on this metric show that in an absolute sense, 87% of personal attacks are at least partially captured by the algorithm. The simplest baseline, which produces rationales by thresholding the coefficients of a logistic regression model, does deceptively well on this metric by only identifying attacking words like "jerk" and "a**hole", but its poor tokenwise performance shows that it doesn't mimic human highlighting very well.

2.4.3 Original model tokenwise recall

A perplexing result of the rationale performance comparison is how good the tokenwise recall of the model is *without* the inverse term. Without it, the model is encouraged to find the minimal rationale which offers good predictive performance. Comments with more than one personal attack (e.g. Figure 2.1) constitute 29% of those with at least one attack and 13% of all comments in our validation set. For comments like these, the model should in theory only identify one such attack. However, it tends to find more information than needed, leading to a higher-than-expected recall of .52 in the best overall version of this variant.

To explain this behavior, we run a leave-one-out experiment on the original+bias and original+bias+inverse model variants. For each distinct contiguous rationale chunk predicted by each model (when it generates multi-piece rationales), we try

removing this chunk from the predicted rationale, running the prediction layer on the reduced rationale, and seeing whether the result lowers the value of the overall objective function.

For the original+bias model variant, we find that performing this reduction improves the value of the objective function 65% of the time. However, the combined average impact of these reductions on the objective function is to worsen it. What this means is while 65% of distinct phrases discovered by the generator are unnecessary for accurate prediction, the 35% of them that are necessary lead to a major decrease in predictive accuracy.

That is, the generator "hedges its bets" with respect to predictive accuracy by including more information in the rationales than it has to, and experiences a better global optimum as a result. This behavior is less prominent with the inclusion of the inverse term, where the percentage of unnecessary rationale phrases falls to 47%.

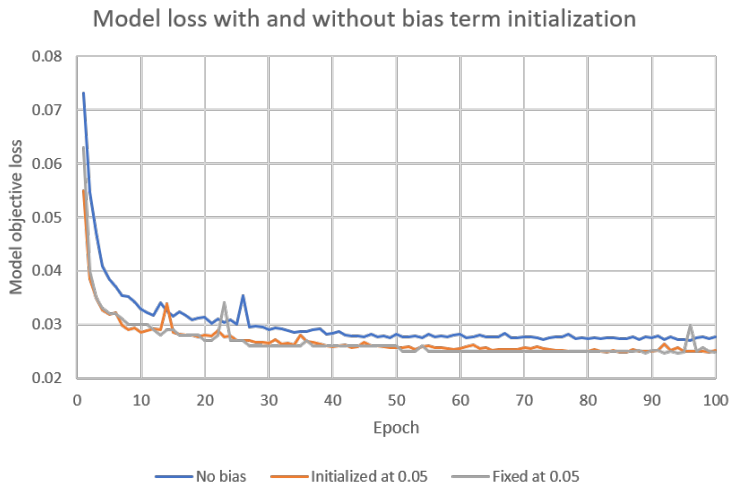

Figure 2.5: Evolution of model loss over time with and without bias term manipulation

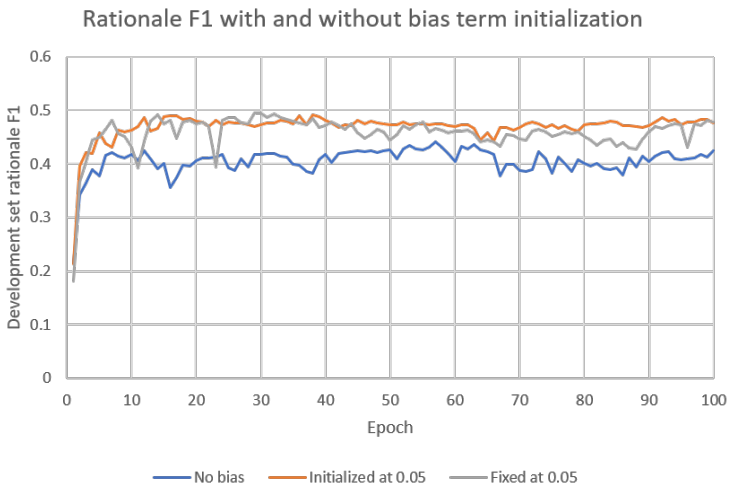

Figure 2.6: Evolution of development set rationale F1 score over time with and without bias term manipulation

2.4.4 Impact of bias term manipulation

In theory, the model should learn a good bias term for the predictor layer, and therefore the idea of explicitly initializing or fixing the bias term to match the semantics of the task should not impact model performance or represent much of a contribution.

In practice however, as Figures 2.5 and 2.6 demonstrate, the initialization of the bias term has a big impact on even the long-term learning behavior of the model. Using the best hyperparameters for the original no-bias, no-inverse-term model, figure 2.5 shows that either initializing or permanently fixing the predictor bias for a default output value of 0.05 leads to improved model loss with respect to its own objective function. Figure 2.6 shows a similar pattern for tokenwise F1 score.

2.5 Discussion

One interpretation of the impact of the bias term on model behavior is that an explanation of "why" is really an explanation of "why not"–that is, an expla-

	True y	Predicted y	Rationalized comment
A)	0.714	0.898	== retard == i ' ts my flippen opinion anyways and how about get a life intead of reporting peoplego get laid you stupid gay virgin
B)	0.964	0.782	== laughable censorship == i see that a post i did in the moon hoax article was ofcourse censored ... fuck you all admins
C)	0.7	0.835	::::: what does that have to do with this article ? take whatever preconceived notions you have about me and shove them up your pussy .
D)	0.727	0.92	` i noticed a tweet by you on the topic of cnn ' s don lemon that read , `` forget about some measure of objectivity as an anchor .`` i was just wondering if you ' re fucking retarded . i mean , you do know you work for fox news , right ? even if we were to limit the discussion to just you , do you think it would be difficult for any person with a working cerebral cortext to find one or two hundred clips of you anchoring on fox and expressing an opinion ? oh , i ' m sorry : not an opinion the opinion of the koch brothers . my mistake . in conclusion , i ' d like to offer you a great big glass of shut - the - fuck - up juice .`
E)	0.292	0.086	== re :== this is a pretty trivial matter , but anyway ... i reverted your `` delightful `` edit because the image you added looked tacky . it is hardly the custom for subject userboxes to contain pictures . and it looks terrible . rest assured , however , that we ' re all just as amazed that you can use latex . wow !
F)	0.1	0.057	i am perfectly within my rights on wikipedia to edit my talk page as i see fit . no user shall edit my page unless they are adding additional , relevant content . irrelevant content shall be removed .
G)	0.3	0.057	` ::: an anarchist who supports `` intellectual property .`` amazing .`
H)	0.3	0.279	` == current picture == can we change it ? maybe one where hes on the middle rope and saying the second `` keeeennnnddeyyyyy `` the current one looks a bit retarded `

Figure 2.7: Further examples of labeled and rationalized comments. Items E) and G) show that the algorithm struggles with sarcasm.

nation is information that distinguishes an item from some alternative hypothesis, and explicitly choosing what this alternative is can improve explanation performance (particularly precision).

Manually setting the model to produce some reasonable default value for an empty rationale makes sense in our setting, but not in domains where there is no default value, such as the beer review dataset of (*Lei et al.*, 2016). A more general approach would be to base explanations on confidence rather than accuracy, where the default value would simply be the mean and variance of the training data, and explanations would consist of tokens that tighten the bounds on the output.

A surprising finding is that the original algorithm often ends up defying its own objective and finds more complete rationales than needed. The leave-one-out experiment described above suggests that the reason for this behavior is that it is how the generator deals with predictive uncertainty, and that it achieves a better global optimum by producing locally suboptimal rationales.

While this "bug" proves useful in our case, it may not generalize. In our setting

the adversarial predictor gives a modest improvement in recall; it will produce a larger improvement in settings where the unaltered algorithm is more successful at producing the minimal explanations described by its objective function. *Li et al.* (2016) finds that a memory network predictor requires less occlusion than an LSTM to flip its predictions, indicating that choice of model can affect completeness of explanations.

In theory, interpretable models can aid human moderators by pointing them directly at the potentially objectionable content in a comment and giving them a starting point for making their own holistic decision about the comment. However, there are potential pitfalls. Adding explanations as a model output gives the model another way to be wrong–one which humans may be even less able to troubleshoot than simple misclassification. Relatedly, explanations may inspire overconfidence in model predictions. Extensive user testing would clearly be needed before any deployment.

2.5.1 Hybrid robustness

The design and initial empirical evaluation of this algorithm proceeds from the assumption that in the domain of toxicity detection, a complete attention mask is one which includes all toxic content. In service to our central goal of improving the robustness of a human/model hybrid, we assume that a human overseer will benefit from as holistic a view as possible of the toxicity of a comment in trying to guess the consensus toxicity of that comment, and we orient our evaluation toward accomplishing this intermediate objective.

However, we have no concrete evidence that high-recall explanations are really optimal explanations in this context. Are high-recall explanations that mimic human highlighting tendencies really optimal for the types of moderating/self-moderating tasks involved in the domain of personal attacks in online social media? This question can only be answered with human subject experimentation, which we address in Chapter III.

CHAPTER III

User Study 1: Effect of Feature Attribution

3.1 Introduction

We performed a user study to evaluate the effectiveness of the adversarial attention algorithm proposed in Chapter II in helping humans make decisions about the toxicity of social media posts.

While the empirical evaluation reported in that chapter measures the algorithm's success in mimicking human effort on the task of identifying personal attacks, we argue in Chapter I that the ultimate goal of interpretable machine learning is to allow human beings to make better decisions about when to trust the predictions of machine learning models.

For the purpose of this study we switch from the task of detecting personal attacks to that of assessing overall toxicity. Both of these targets are dimensions of the dataset introduced by *Wulczyn et al.* (2017), but toxicity represents a more holistic (but more loosely defined) indicator of the overall objectionability of a given social media comment. Therefore, while while we argue that personal attacks were an easier target for annotators in the Chapter II empirical evaluation to label at the token level, toxicity represents a more externally valid target for a user study focused

[0]This chapter consists of content published in
Carton, S., Mei, Q. and Resnick, P. (2020). Attention-Based Explanations Don't Help Humans Detect Misclassifications of Online Toxicity, *In submission.*

on comment-level labels.

The basic structure of the study was as follows: we sampled comments from the *Wulczyn et al.* (2017) dataset representing a range of true toxicity scores and model errors (i.e. comments on which the model was correct and incorrect). In a first phase, we recollected ground-truth toxicity scores for these sampled comments in order to ensure low variance in our outcome measurements. Finally, as a second phase we conducted a 2×2 between-subject experiment that assessed the impact of adding 1) a model prediction as a visual element alongside the comment text, and 2) explanations for the predictions of that model via highlighting relevant words and phrases within the text. For clarity, we refer to participants in the first phase as "workers" and those in the second phase as "subjects".

We also included two extension conditions that tested variant explanation techniques; a "partial" variant that highlights a minimal amount of relevant text, and a "keyword" variant that only identifies toxic words without regard for context or phrase structure. Figure 3.1 summarizes the 6 experimental groups.

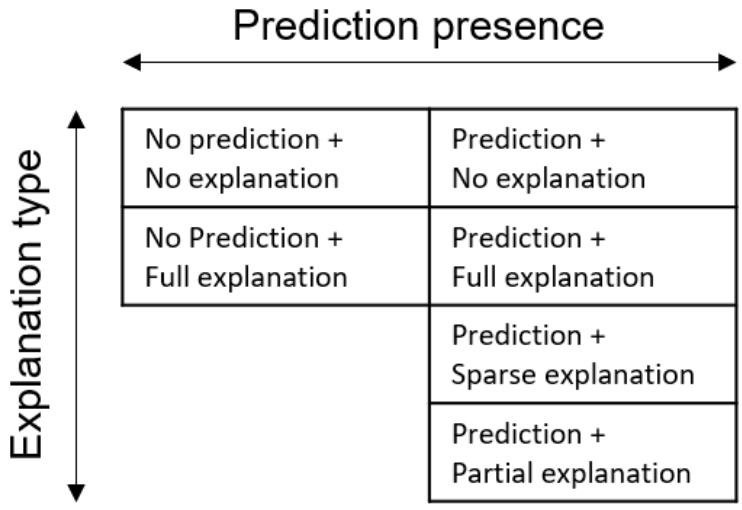

Figure 3.1: Experimental conditions.

This user study was designed to investigate three research questions:

RQ1: Presence of model predictions. How does the advice of a partially informative but unreliable predictive model affect subject performance?

RQ2: Presence of explanations. Do (attention-based) explanations help subjects make better use of advice from an unreliable model in predicting the perceived toxicity of social media comments?

RQ3: Explanation type. Do more minimal "partial" or sparser "keyword" explanations exhibit different performance properties from explanations optimized for completeness?

We used the adversarial attention model described in section II to generate attribution masks on sampled comments. We generated the "partial" variant by reducing each mask to only the single worst instance of toxicity in the text. We generated the "sparse" variant by applying essentially a dictionary approach, identifying individual words associated with toxicity according to the coefficients of an independently trained logistic regression model.

The contributions of this chapter are as follows:

- We test the feasibility of interpretable machine learning for semi-automated toxicity detection.
- We add to a small but growing body of evidence suggesting that the most popular types of explanations aren't adequate to improving human performance on decision tasks.
- We test the relative effectiveness of three different approaches for extractive, feature-based explanations of text classifier decisions.

3.2 Experiment Design

The experiment sought to evaluate how well subjects predict the perceived toxicity of social media comments with varying levels of algorithmic assistance. It consisted of the following protocol:

1. **Sample comments:** Draw a sample of comments from the *Wulczyn et al.* (2017) dataset, selecting for diversity in toxicity scores and model error. 96 comments sampled, split into 2 sets of 48 each. Each subject labels one comment set.

2. **Model comments:** Train attention model and run it on all 96 comments, producing predicted toxicity score and attention mask for each comment. Different subject groups label comment sets with different combinations of these features visible.

3. **Collect ground truth (phase 1):** Collect low-variance ground truth toxicity score for each comment by asking workers for their personal opinion of each comment and aggregating response. 54 subjects reviewed each of the 2 comment set, 108 workers total.

4. **Predict ground truth (phase 2):** Ask subjects to predict outcome of phase 1 with varying levels of algorithmic assistance. 40 subjects reviewed each of the 2 comment sets across 6 treatment conditions, 480 subjects total.

The structure of the phase 2 experiment was a 2×2 between-subject design with two treatments: presence of prediction and presence of explanation, as well as two extension conditions in which the prediction is present with a variant explanation type, "keyword" and "partial" (Figure 3.1).

3.2.1 Subjects

Subjects were recruited using the Amazon Mechanical Turk platform in August 2018. Subjects had to be US-based, and had to have completed at least 1000 HITs with 95% acceptance or more in order to qualify for the experiment. The study involved 588 total participants, 108 workers in the phase 1 labeling task and 480 subjects who in the phase 2 experiment (as enumerated above).

This subject count was chosen through a simulated power analysis to have a high (80%) chance of detecting an effect size of 0.05 in the primary outcome, accuracy, given outcome variances observed in a pilot study. This minimum detectable effect size was chosen as representing a 10% improvement on what was observed to be the baseline human accuracy of roughly 50% on the task. Details on outcomes and statistical analysis are given below.

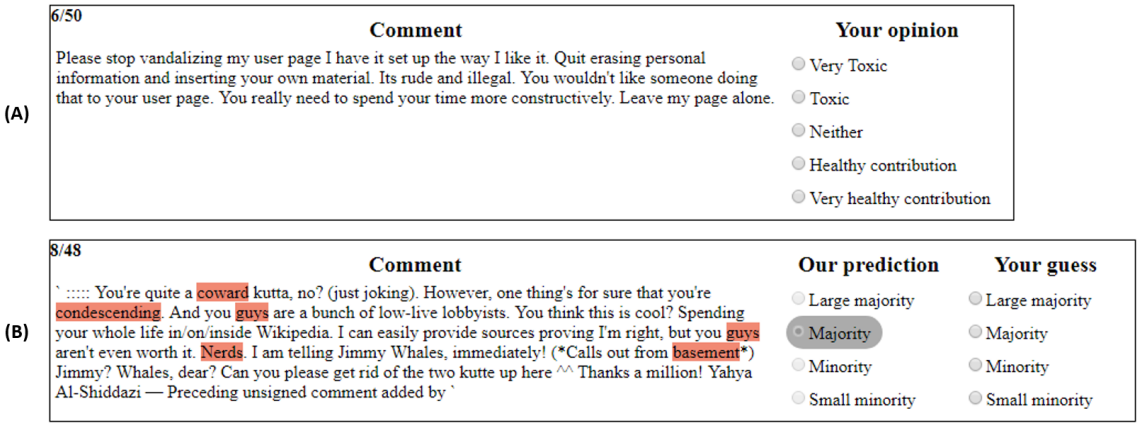

Figure 3.2: (A) Example comment in the phase 1 personal opinion task; (B) Example comment in the phase 2 prediction task

3.2.2 Comment sampling

We sampled comments from the *Wulczyn et al.* (2017) dataset. This dataset consists of roughly 100,000 Wikipedia revision comments each labeled on a 5-point

toxicity scale (figure 3.2A) by at least 10 workers on the CrowdFlower platform. We followed *Wulczyn et al.* (2017) in binarizing each 5-point label to toxic(1)/nontoxic(0), and took the fraction of users who found the comment toxic to form a continuous toxicity value for each comment. Hence, a comment which 3/10 CrowdFlower workers deemed toxic is assigned a 0.3 true toxicity label for the purpose of model training and evaluation.

For the purpose of the experiment, we convert this toxicity prediction task into a four-class classification task, with each class representing one quartile of the true toxicity score: Large majority (75% to 100%); Majority (50% to 75%); Minority (25% to 50%); and Small minority (0% to 25%). We chose to frame the task this way rather than as a binary classification task in order to make it more difficult for human participants, and therefore to provide more room for improvement in accuracy.

The *Wulczyn et al.* (2017) dataset is quite unbalanced. Roughly 90% of instances have a toxicity score below 0.5. Furthermore, it represents a relatively "easy" classification task: our LSTM classifier achieves 96% accuracy against the (binarized) true toxicity scores. We were interested in understanding human performance across the full range of true labels. Furthermore, we wanted to investigate whether explanations could allow human users to overturn classifier errors. Hence, in choosing which comments to present to our human subjects, it was necessary to perform stratified sampling on these two qualities: true toxicity and model error.

Specifically, we sampled 48 comments total for each comment set, split evenly across the 4 toxicity quartiles described above. For each quartile, we sampled 12 comments: 6 where our model predicted the correct quartile, and 2 each of where the model predicted each of the 3 other quartiles. For the two edge quartiles, large majority and small minority, there were not enough cases where the model predicted the other extreme. For these, we instead sampled 3 from the next most extreme error and only 1 from the most extreme.

Put together, this process resulted in a sample which is 50% toxic/nontoxic, 25% in each quartile, and on which our model achieves 50% classification accuracy at the quartile level (with respect to the labels present in the *Wulczyn et al.* (2017) dataset). Thus, subjects were presented with a roughly even number of comments that were toxic versus nontoxic, and a roughly even number for which the classifier was correct versus incorrect.

As a result of this process, we presented participants with a sample of comments on which the model is quite inaccurate. As we discuss further below, it is in fact less accurate than the baseline accuracy of human subjects in our study. This is a contrast to similar studies such as *Poursabzi-Sangdeh et al.* (2018) and *Lai and Tan* (2019), where the model was more accurate than the human subjects. In those studies, a big improvement to human accuracy was possible simply by persuading subjects to agree with the model, so an effective explanation was one which increased user trust in the model output. In our study, no such avenue existed.

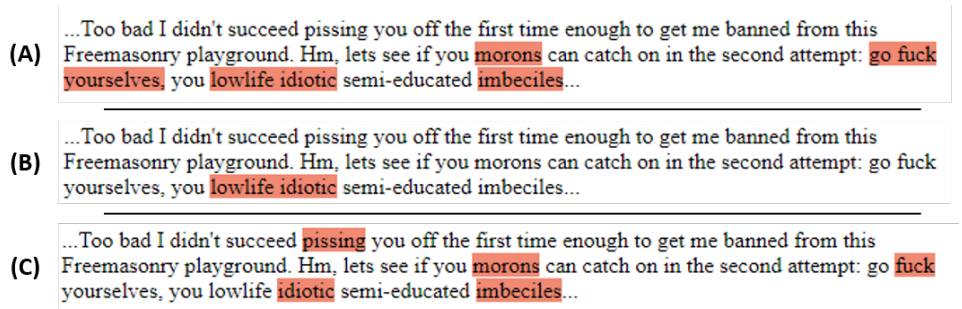

Figure 3.3: Example of explanation variants: (A) Full explanation; (B) Partial explanation; (C) Keyword explanation

3.2.3 Modeling

For every comment, we generated both a prediction about the toxicity of that comment and a feature attribution mask indicating which tokens in the comment the model thought were toxic (Figure 3.2B). We use the model described in section 2.2

to generate these predictions and masks.

We supplemented the full model with two variants which were intended to gauge the effectiveness of different styles of feature attribution for text.

In the first variant, we produced "partial" explanations by taking any multi-phrase explanation produced by the model and reducing it to just the single phrase which maximizes the accuracy of the predictor when considering only that phrase. So when a comment has multiple discrete instances of toxicity, we reduce the explanation to just the most toxic instance (see figure 3.3B). This variant was intended to test the hypothesis that an explanation needs to consist only of the most decisive information present in an instance rather than all relevant information.

The second variant produces keyword-based explanations. We train a bag-of-words logistic regression classifier on the same dataset as the full model, and use the coefficients of this model to designate certain words as toxic (e.g. "pissing", "morons" in figure 3.3C). This amounts to a dictionary-based approach, where certain words are always considered toxic and others nontoxic. It produces very sparse explanations, where only the most toxic single words are highlighted, without regard for context or phrase structure. This variant was intended to test how important it is to capture whole phrases, or whether identifying individual words is sufficient.

3.2.4 Ground truth collection (phase 1)

In phase 1, we recollected ground-truth toxicity scores despite having access to an existing ground truth in the *Wulczyn et al.* (2017) dataset. We did so by having 54 subjects label each comment using the same questionnaire as *Wulczyn et al.* (2017), which asks the worker to rate the comment on a 5-point scale between "Very toxic" and "Very healthy" (Figure 3.2A). When we aggregated the results of this phase, we binarized each response into either toxic ("toxic" or "very toxic") or nontoxic (any other option), took the mean across subjects, and then bucketed each mean into the

appropriate quartile to serve as the true toxicity label for that comment.

We recollected these labels for several reasons. First, having 54 subjects for each comment instead of 10 meant a generally lower-variance true label for each comment. Second, drawing our ground truth from the same population as the phase 2 subjects was more fair to them, since that phase involved asking them to make predictions about their own population rather than that of CrowdFlower.

The third reason is somewhat more nuanced. As a subjective quality, the true toxicity of a comment is a distribution, not a point value. Any attempt to estimate the mean of this distribution by surveying a population is going to have a certain amount of random error, where for some items a significant fraction of labelers are bots, trolling, inattentive, etc. Because we sampled a disproportionate fraction of items where the classifier was incorrect, we were worried that a disproportionate number of these items would be ones where the label itself was noisy due to random labeler error. The phase 1 experiment, therefore, served to reduced this chance by re-surveying the toxicity of the selected items.

As for why we chose to follow the *Wulczyn et al.* (2017) questionnaire in the first place and define ground truth toxicity scores as a mean of binary responses, the reason for this is synchronicity with the model. If we recollected a ground truth generated differently from this dataset (and therefor drawn from a different distribution), the model's predictions and explanations would be tuned to a different data distribution than this ground truth, and this disjunction would represent a threat to the to the validity of the study.

3.2.4.1 Phase 1 quality assurance and compensation

Quality assurance for phase 1 was via two attention checks in each question set. Subjects were made aware of the presence of the attention checks, though not of how many there were. Each attention check consisted of a sentence embedded within a

comment asking the user to assign it a certain label chosen to be the opposite of the true label for that item. Workers thus were likely to miss the attention checks if they were putting random labels or failing to carefully read the comment texts.

Phase 1 workers were compensated with a base payment of $1.50 plus a bonus of $0.50 for each attention check they marked correctly. We discarded the results of any subject who missed both attention checks (3 in total).

3.2.5 Prediction experiment (phase 2)

In phase 2, we asked subjects to predict the outcome of phase 1. Hence, if a comment was designated toxic by 60 % of the subjects who reviewed it in phase 1, the target class for that comment would have been "majority" in phase 2.

The purpose of phase 2 was to examine how well subjects were able to integrate advice from an unreliable model into their own predictions, and the extent to which explanations made them more or less effective in doing do.

As described briefly in previous sections, each phase 2 subject made predictions under one of six different experimental conditions:

1. No prediction + no explanation (control)
2. Prediction + no explanation
3. No prediction + full explanation
4. Prediction + full explanation
5. Prediction + partial explanation
6. Prediction + keyword explanation

Phase 2 subjects were asked to review each text and choose one of the toxicity quartiles described above (figure 3.2B).

In the control condition, workers made toxicity predictions without any algorithmic assistance. Two treatments were explored: the presence of the algorithmic predictions, and the presence of explanations in the form of word highlighting As

described above, explanations came in three variants: full, partial and keyword-based (figure 3.3).

In prediction-present conditions, the algorithm's prediction was presented to the right of the comment text figure 3.2B). In order to prevent workers from simply mirroring the model prediction, the instructions explained that the model was "not entirely reliable", and that workers would have to decide how much they wanted to rely on it.

In explanation-present conditions, the explanation was presented as red highlighting over the comment text (also figure 3.2B)). This feature was explained to users as the algorithm attempting to highlight toxic content.

3.2.6 Phase 2 quality assurance and compensation

Workers in phase 2 were given a base payment of $1.25 plus a bonus of $0.05 for each item they predicted correctly relative to the aggregated results of phase 1. We didn't use any other quality assurance mechanism for two reasons. First, we were relying on the natural desire of our subjects to maximize their earnings under the stipulation of the unreliable model. Second, we wanted to observe a natural distribution of carelessness–forcing subjects to read every comment carefully would have been unrealistic compared to how social media posts are consumed in a real-world setting.

3.3 Results

We consider the results of the phase 2 experiment in terms of 5 outcome variables which cover the accuracy, speed and trust that subjects felt in the classifier:

- Quartile prediction accuracy
- Quartile prediction false positive rate
- Quartile prediction false negative rate

- Agreement with classifier prediction
- Median seconds-spent-per-comment

We achieved a binary Krippendorf's Alpha of 0.51 on the phase 1 experiment, comparable that of 0.45 reported in *Wulczyn et al.* (2017).

In our analysis of phase 2 results, we calculate the effect size of each condition in comparison the most appropriate control for that condition given our research questions. To assess the impact of predictions and explanations alone (RQ1), we compare "No prediction + full explanation" and "Prediction + no explanation" against "No prediction + no explanation". To assess the impact of explanations given the presence of a prediction (RQ2) we compare "Prediction + full explanation" against "Prediction + no explanation". Finally, to understand the relative impact of the two explanation variants (RQ3), we compare both "Prediction + partial explanation" and "Prediction + sparse explanation" against "Prediction + full explanation".

For every comparison we perform a two-tailed t-test. We report the p-value for each comparison, adjusted by Benjamini-Hochberg correction across the 5 comparisons and 5 outcomes with a target false discovery rate of 0.05.

3.3.1 Accuracy and agreement

	Condition	Mean % highlighted	Accuracy		Agreement	
			Mean	p	Mean	p
	Model		0.375		1	
1	No prediction + no explanation		0.544		0.432	
2	No prediction + full explanation	0.26	0.514	0.2938^{1}	0.436	0.9585^{1}
3	Prediction + no explanation		0.525	0.5129^{1}	0.535	0.0000^{1***}
4	Prediction + full explanation	0.26	0.524	0.9585^{3}	0.533	0.9585^{3}
5	Prediction + partial explanation	0.234	0.526	0.9585^{4}	0.519	0.6651^{4}
6	Prediction + keyword explanation	0.048	0.518	0.9585^{4}	0.531	0.9585^{4}

Table 3.1: Mean subject quartile accuracy, agreement with model, and percentage of text highlighted across conditions. p-value superscripts indicate comparison condition.

Table 3.1 summarizes the mean quartile accuracy of users in each condition, as well as that of the model. It also summarizes the mean agreement of subjects in each condition with the model's predictions, as well as the mean percentage of tokens that are highlighted by each attribution method.

We find that the presence of the model's prediction has a marginal negative effect on the accuracy of subjects, an effect that does not vary significantly with the presence of explanations (of any variant). However, we do find that the presence of a visible prediction significantly increases subject agreement with the model, though explanations do little to moderate this interaction (Figure 3.4).

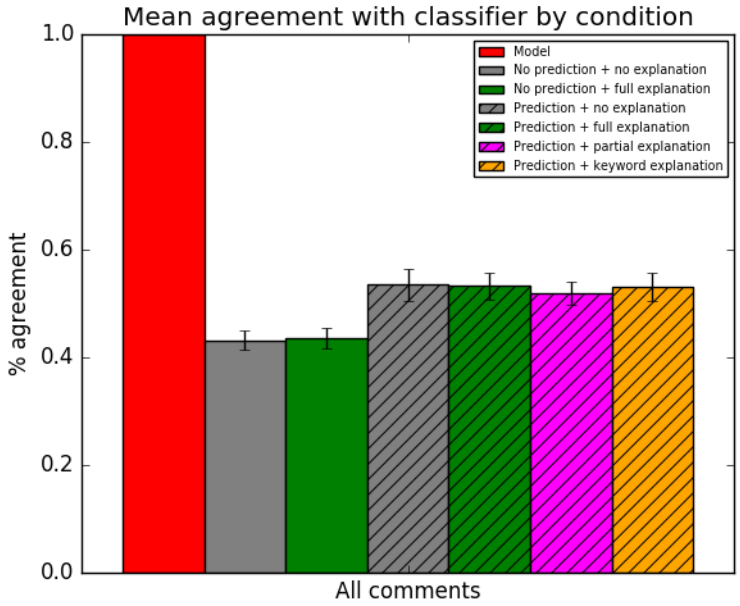

Figure 3.4: Mean agreement of users with model across experimental conditions and question subsets with 95% confidence intervals.

This agreement effect explains the accuracy effect of prediction presence. As figure 3.5 shows, when the model is correct, the presence of predictions increases subject accuracy. When the model is incorrect, it decreases accuracy. Because the model is (by design) relatively inaccurate in this study, this leads to slightly more cases of the model negatively influencing subjects than positively influencing them.

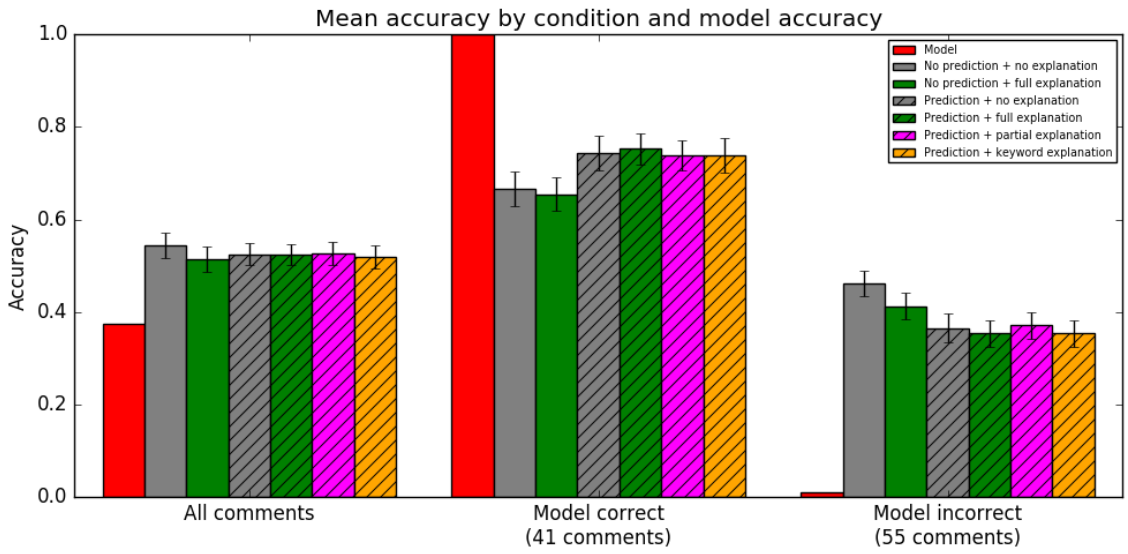

Figure 3.5: Mean quartile accuracy of model and users across experimental conditions and model correctness.

3.3.2 False positive rate and false negative rate

	Condition	False negative rate		False positive rate	
		Mean	p	Mean	p
	Model	0.396		0.229	
1	No prediction + no explanation	0.276		0.179	
2	No prediction + full explanation	0.353	0.0042^{1*}	0.133	0.0338^{1*}
3	Prediction + no explanation	0.315	0.1543^{1}	0.16	0.3891^{1}
4	Prediction + full explanation	0.337	0.3891^{3}	0.139	0.2938^{3}
5	Prediction + partial explanation	0.357	0.4344^{4}	0.116	0.2892^{4}
6	Prediction + keyword explanation	0.346	0.924^{4}	0.135	0.9585^{4}

Table 3.2: Mean false positive rate and false negative rate across conditions. p-value superscripts indicate comparison condition.

While we find no significant effect of explanations on accuracy per se, breaking subjects errors down into false negatives and false positives shows they do impact the distribution of errors made by humans (Table 3.2).

In particular, we find that explanations alone increase false negative rates while decreasing false positive rates relative to the completely unassisted condition (Figures

3.6 and 3.7). This result implies that feature attribution changes the way that subjects read the comments, making it easier for them to avoid errors of attribution but more liable to make errors of omission

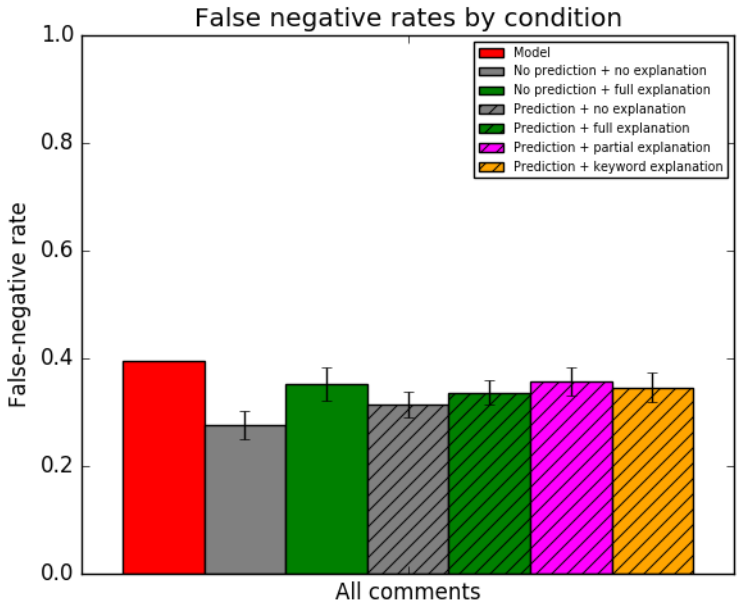

Figure 3.6: Mean false positive rate of subjects across conditions.

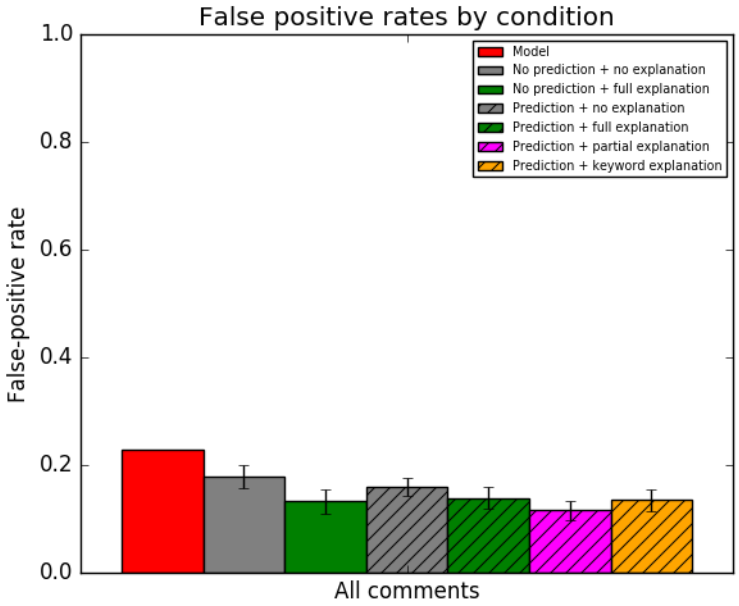

Figure 3.7: Mean false negative rate of subjects across conditions.

3.3.3 Speed

	Condition	Seconds/ comment	
		Mean	p
1	No prediction + no explanation	10.162	
2	No prediction + full explanation	9.752	0.5772^{1}
3	Prediction + no explanation	11.869	0.0445^{1*}
4	Prediction + full explanation	9.954	0.0320^{3*}
5	Prediction + partial explanation	10.645	0.3735^{4}
6	Prediction + keyword explanation	9.878	0.9196^{4}

Table 3.3: Mean seconds-per-comment across conditions. p-value superscripts indicate comparison condition.

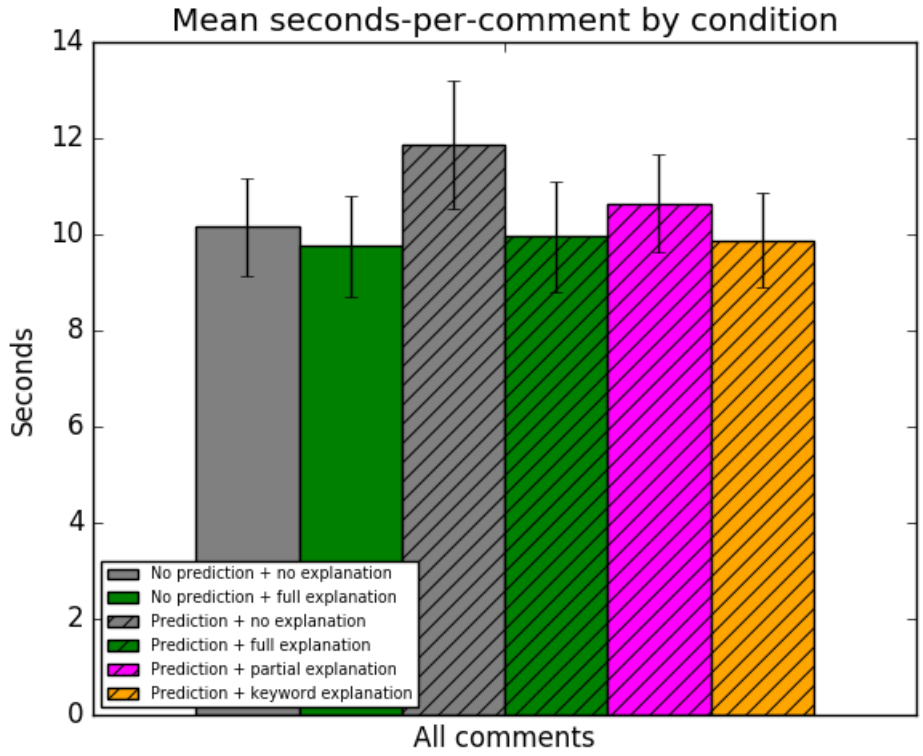

Figure 3.8: Mean seconds-per-comment of subjects across conditions

Table 3.3 summarizes the speed effect of the various conditions. We find that the addition of a prediction adds a significant time penalty in comparison to the unassisted condition, presumably as users are forced to attempt to reconcile their

own opinions with that of the classifier. However, adding explanations erases this time penalty, bringing the mean comment labeling time back down to that of the unassisted condition. Figure 3.8 demonstrates this visually.

3.4 Discussion

The results described above provide answers to our five research questions:

3.4.1 RQ1: Presence of model predictions

We find that the presence of a visible model prediction tends to bias subjects in favor of the prediction, whether it is correct or incorrect. There is also a significant speed penalty associated with the presence of a model prediction, as users are forced to ingest and reconcile an additional piece of information beyond the text itself.

This suggests that it is difficult for users to effectively integrate advice from a model into their decision-making. In our experiment, they are as likely to discard good advice as to reject bad advice.

3.4.2 RQ2: Presence of explanations

We find no significant effect of explanations on user accuracy or agreement when exposed to a model prediction. One possible explanation for this failure is that, in this domain, the difficulty in prediction lies not in identifying what words and phrases may be toxic, but in predicting exactly how those words and phrases are liable to be perceived by the general population.

Dividing subject errors into false negatives and false positives sheds a bit more light on the situation. The model has both a high false negative rate and false positive rate compared to unassisted subjects, but both prediction and explanations alone raise the false negative rate and *lower* the false positive rate among human

subjects. Explanations alone lower the FPR by a greater amount than predictions alone.

This result suggest that in this context, explanations are particularly liable to cause subjects to make mistakes of omission, presumably as they focus only on the text that has been highlighted without seriously considering the un-highlighted text (which may sometimes contain evidence of toxicity).

The one unequivocal benefit we do find is that explanations erase the speed penalty of prediction presence, allowing users to more speedily determine whether they believe or disbelieve in the model output.

3.4.3 RQ3: Explanation type

We find that "partial" explanations, which identify only a single instance of toxicity, cause subjects to produce more marginally more false negatives relative to "full" explanations optimized to catch all toxicity. This suggests that it is important to explicitly optimize for completeness, possibly because human subjects find incomplete explanation confusing and contradictory, and so are forced to dedicate more time to understanding them.

By contrast, we find no significant difference between the full explanation model and the much simpler "keyword" explanation variant which just highlights potentially problematic words regardless of context, even though the "keyword" variant occludes 5 times less content on average than the "full" variant (Table 3.2). This results suggests that while it is important for explanations to reflect all toxic material in a given comment, it is less important that they cover entire phrases.

3.4.4 Experiment design

The results of this study, in combination with other similar recent studies like *Lai and Tan* (2019) and *Nguyen* (2018), seem to imply that relative balance of model and

human skillfulness is a crucial factor in the design of evaluation studies for explanatory machine learning.

In order for explainable machine learning to be useful from a decision quality perspective, we argue that they have to allow human operators to make more accurate decisions than either unassisted human baseline accuracy or unsupervised model accuracy. Otherwise, there is no point in combining the two types of agent–one or the other working alone would be a better solution.

For this to be the case, there need to be a substantial proportion of instances for which model performance is good *and* human performance poor, and vice versa. *Kleinberg et al.* (2017) found this to be the case for recidivism prediction, as an example. The more of a performance gap exists between baseline human and model performance, the less common such instances will be, meaning that the greatest potential improvement exists when human and model baseline performance is roughly equal.

To achieve such a balance in this experiment, we generated a sample of comments from the *Wulczyn et al.* (2017) stratified by model error relative to the existing ground truth, and we were careful to include one experimental condition for assessing baseline human accuracy. While we did not observe an accuracy effect in our study, we believe that failing to account for these issues in future studies may result in spurious effects.

Another important point of experimental design explored in this study is that of re-collecting ground truth. There is both a variance and bias argument for doing this. The variance argument is that for a task based on consensus human labels such as toxicity or sentiment, individual ground truth labels may be too noisy for an accuracy effect to be recognizable in a medium-scale human user study. The bias argument is that if items are sampled in a way which is dependent on their existing ground-truth labels (e.g. as described above) this selection process may result in sampled items whose noise distribution is different from the dataset as a whole. Recollecting ground

truth labels for items can mitigate both of these potential issues.

3.4.5 Limitations

The experiment had several limitations that will have to be addressed in further work. First, the three explanation variants we test are not fully representative of the current interpretability literature. Rather, they represent three extremes: capturing *all* locally pertinent information (full variant), capturing *minimal* locally pertinent information (partial variant), and capturing independent globally pertinent information (keyword variant). It is possible that there exists some feature highlighting technique (e.g. *Arras et al.* (2017)) that would produce better outcomes, though it seems unlikely given the relatively consistent negative result across the three explanation variants as well as other similar studies.

We also limited ourselves to discrete binary highlighting–a token is either in or out of an explanation, without further embellishment. We did not include words and phrases of nontoxic valence, nor did we allow for grades of relevance, as in *Arras et al.* (2017). It is possible that a more informative style of feature-highlighting would produce the accuracy benefits that we failed to observe in this work.

We display all information at once–that is, text, prediction and highlighting were all presented together to each user. A multi-phase presentation, where users are prompted for an initial decision before being exposed to any algorithmic assistance, might result in less bias toward the model prediction. However, it would also reduce the potential for time savings, as users would have to go to all the trouble of making a careful decision before getting a chance to process the output of the algorithm.

Finally, the stratified question sets we employed in this experiment are significantly more toxic than a random sample of social media comments would be, while the model was significantly less accurate than a model would be on randomly distributed data (40% versus the 96%). While we were able break down task performance

by individual-comment accuracy, the particular distribution of toxicity and classifier error probably prompted subjects to be more skeptical of the model and differently sensitive to toxicity than if the comments had been sampled in a more representative manner.

3.4.6 Toxicity detection versus moderation

Our experiment involves untrained Mechanical Turk workers making predictions about what percentage of other workers are likely to find comments toxic. The comments they view represent a variety of different true levels of toxicity and are removed from conversational context. This is somewhat abstracted from a true moderation setting, where trained moderators apply a specific set of community standards to comments, typically in response to some kind of reporting mechanism.

However, the purpose of this study is less to prototype a machine-assisted moderation system than to test the impact of interpretable machine learning on human performance on a decision task that involves a tension between existing intuitions and an external standard for correctness. In a true moderation task, the external standard would consist of a set of community guidelines; in our experiment it is the consensus label established by the phase 1 labeling task.

The question of the difference between "toxicity detection" and moderation is an important one, but it is also one that belongs to the larger literature on machine approaches to online abuse. The Perspective API, for example, is a prominent and well-cited service for assessing the toxicity of social media posts, trained on the same dataset used in this paper. While there has been criticism of the idea of classification in place of human moderation (e.g. *Blackwell et al.* (2018b)), we are not aware of existing theoretically-motivated attempts to reconcile the classification task encoded in a dataset such as *Wulczyn et al.* (2017) with the task of moderation as experienced

[0]https://www.perspectiveapi.com

by real-world moderators.

3.4.7 Design implications

This study has several implications for systems which seek to provide advice from a text classifier to a human worker, particularly on a subjective task such as toxicity detection.

The way in which explanations change the distribution of human error suggests that a system builder needs to be very careful in their choice of explanation mechanism, because explanations could exacerbate a pre-existing tendency toward false negatives. If time is not a factor, it may actually be better in some cases to have no explanatory mechanism, as this forces users to be thorough in resolving any disagreement between themselves and the advisory classifier. The good news is that the lack of difference between the full and keyword-based explanation types suggest that simple highlighting methods, even dictionary methods, can be just as effective as more sophisticated ones, as long as they are tuned to catch as much relevant content as possible.

This need for explanation completeness seems particularly true for an intuitive task such as toxicity detection where subjects are more able to troubleshoot errors of spurious association than errors of omission. This is a result that designers of semi-automated moderation systems would need to consider particularly carefully.

Finally, the high-level implication of this study is that interpretable machine learning is not necessarily the immediate panacea to unreliable models that the interpretability literature tends to assume it is–the most popular type of explanation fails to improve how well humans use such a model, and poor explanations can actually reduce human performance by discouraging critical thinking about the model's predictions.

3.4.8 Hybrid robustness

This study represents one attempt to achieve the hybrid robustness improvement that is the central goal of this dissertation. While it achieves several intriguing secondary outcomes, it does not succeed in improving human predictive performance on a decision task with advice from a predictive model. While it is possible that this a result of the particular algorithm we use in this task (the one described in Chapter II), the relatively unanimous results of similar studies like *Lai and Tan* (2019), *Lage et al.* (2018) and *Weerts et al.* (2019) demonstrate that this improvement is a very difficult outcome to achieve.

One possible reason for this consistent failure is that feature-based explanations are not enough to improve human performance. Simply clarifying which features the model attended to and (in some systems) what impact they had on the output may not help humans make better decisions in domains where the decision itself is ambiguous or difficult. For example, it might be relatively easy (and therefore unhelpful) for a human to identify "I hate you" as a potentially toxic phrase within a larger text, but less easy for them to classify exactly what percentage of people are likely to find that phrase toxic.

Hence, in Chapters IV and V we turn to example-based explanations as a potential solution to this shortfall. Where a feature-based explanation is limited in how much insight it can shed on an ambiguous phrase like "I hate you", an example-based explanation can actually reach into the dataset to find other instances of that phrase, and present those examples as evidence for a given outcome.

CHAPTER IV

Attribution-Conscious Explanatory Examples

4.1 Introduction

Most of the explanatory machine learning methods that have been proposed in recent years have focused on clarifying the relationship between model input and output. Feature attribution methods such as deepLIFT (*Shrikumar et al.*, 2017), layerwise relevance propagation (*Bach et al.*, 2015) and LIME (*Ribeiro et al.*, 2016) are the most common instances of this approach, but even other types of techniques such as rule-based explanations (e.g. *Lakkaraju et al.* (2016)) have this basic goal.

The swell of technical work in this area has been accompanied by a corresponding swell of experimental work seeking to understand whether and to what extent explanation methods can improve human understanding and trust in machine learning models (e.g. *Poursabzi-Sangdeh et al.* (2018); *Lai and Tan* (2019); *Narayanan et al.* (2018)). What these studies tend to have in common is that they generally observe either no impact or a marginal impact of explanations on the accuracy of the decision ultimately made by the human user. That is, it remains unclear how (and if) explanatory machine learning can lead to more accurate predictions from hybrid human-machine systems.

[0]This chapter and the following one consist of content published in
Carton, S., Mei, Q. and Resnick, P. (2020). Model Attention for Example-Based Explanations of Text Classifiers, *In submission.*

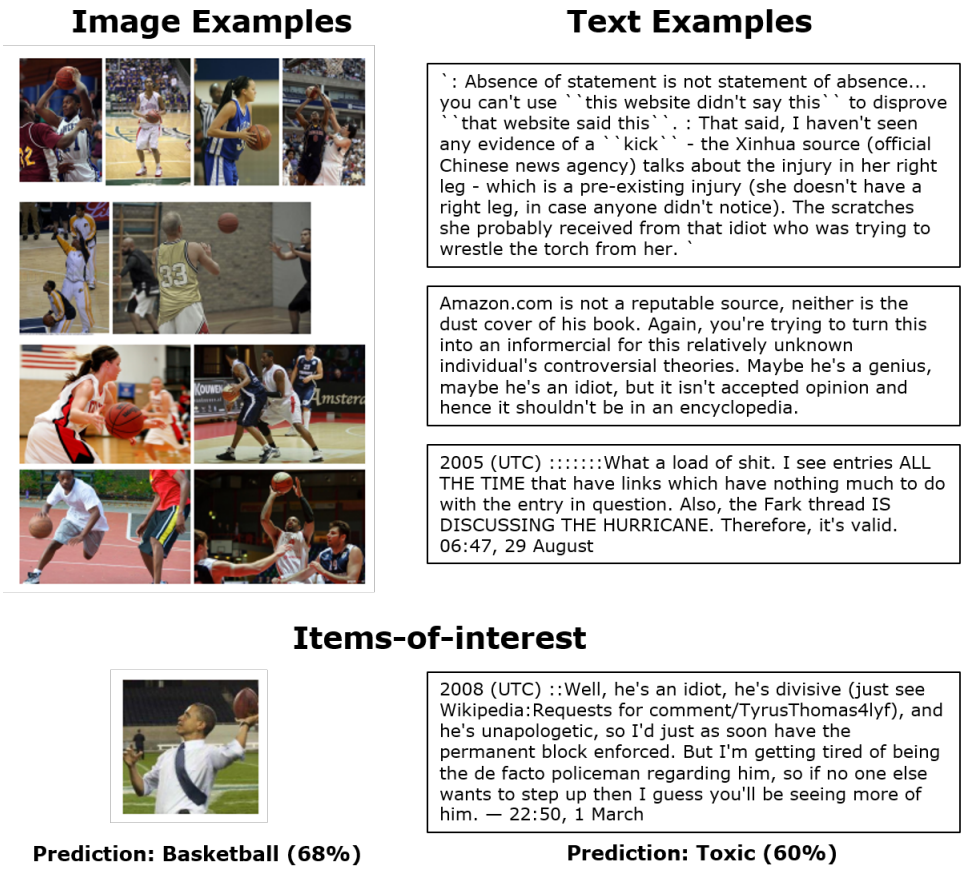

Figure 4.1: Image versus text examples. Without additional visual cues, it is difficult to assess text similarity. Image examples courtesy of (*Papernot and McDaniel*, 2018).

This would seem to suggest that a different basic approach is needed. In many cases it may simply not be possible to improve human accuracy by explaining the relationship between model input and output. In the case of predicting social media comment toxicity, it may be that no articulation of what parts of a text the model considers toxic, or how the model would respond to various types of perturbations of the input, can give a human being insight into what percentage of people would find a phrase such as "this idea is stupid" to be toxic.

Rather than just clarifying the relationship between model input and output, it may be necessary to turn to example-based explanations, which extract or generate evidence from the training data in order to justify the model's predictions (*Guidotti*

et al., 2018). What this type of explanation offers beyond any feature-based explanation is the opportunity to perform analogical reasoning. Given an item-of-interest x^i and the model's prediction about it $f(x^i) = \hat{y}^i$, explanatory examples can give a user a way to reason about the reliability of that prediction based on the model's behavior (and performance) on similar items.

For the purpose of this work, we consider explanatory examples to be real $(x, \hat{y}, y)$ triplets drawn from the training set. Every algorithm we consider for extracting these examples uses the same basic two steps: 1) develop a compact "retrieval representation" of each item-of-interest which preserves useful semantic qualities; then 2) retrieve explanatory examples by performing a nearest-neighbor search within that representation space. For this reason we use the terms "explanatory example" and "neighbor" interchangeably in this work.

The task of generating explanatory examples for text classification presents different challenges than for image classification, which is where most recent literature has focused (e.g. *Kim et al.* (2016)). One obvious difference is that it is much more difficult for humans to make quick comparisons between texts than between images (Figure 4.1). In fact, unassisted comparison of texts represents such a cognitive burden that we argue it represents the first major hurdle in allowing this style of explanation to be practical for text data.

4.1.1 Feature attribution, relevance and fidelity

Feature attribution can help reduce the cognitive burden of comparing texts. By reducing each text to only the tokens that were impactful in the model's prediction, feature attribution can clarify how the explanatory examples are related to the comment of interest (Figure 4.2).

However, performing this reduction can reveal issues that threaten the analogical validity of these examples. For example, Figure 4.2.2, while it is of similar predicted

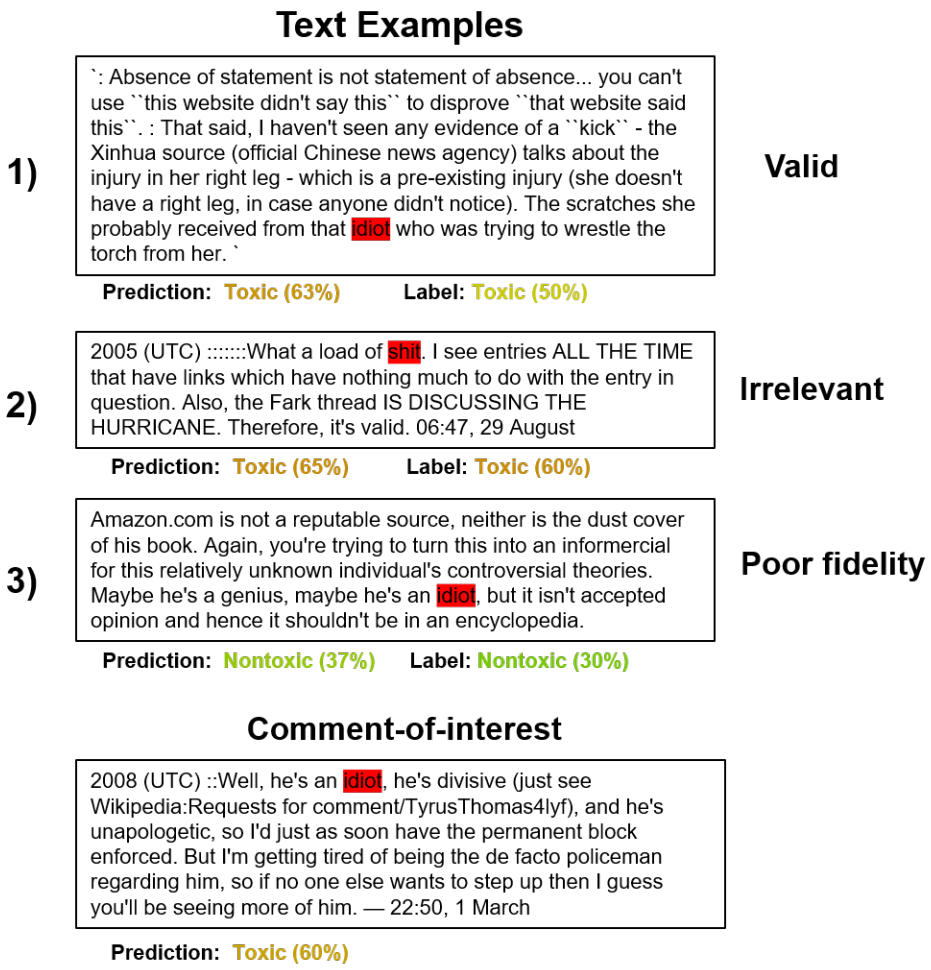

Figure 4.2: Text examples with feature attribution. Neighbor 1 is a valid analogy for the item of interest; Neighbor 2 is irrelevant; Neighbor 3 is visually similar but displays poor fidelity with the model's decision on the item of interest.

toxicity to the comment-of-interest, uses different vocabulary. A human user would hesitate to draw conclusions about the true toxicity of the comment-of-interest from such an irrelevant example.

By contrast, Figure 4.2.3 has very similar attributed content, but its predicted toxicity is different from that of of the comment-of-interest. As a result of this infidelity to the model's behavior on the item-of-interest, it would be problematic to use that particular example as an indicator about how accurate the model is likely to be on the comment-of interest.

An important distinction to make here is between *shallow relevance* and *deep relevance*. When the algorithm identifies a neighbor such as Figure 4.2.1 with similar attributed content and a similar prediction to the item-of-interest, it is clear why the algorithm considers the two to be similar. However, a human overseer with a deeper semantic understanding of the text may be able to realize that the neighbor is different in meaning from the item-of-interest despite its superficial similarity.

That, we propose, is the point at which human and machine expertise can interact in this type of explanation. The retrieval algorithm can identify neighbors of superficial (shallow) relevance, and the human can select from among these the neighbor with the greatest real (deep) relevance to use as a precedent for their decision about the item-of-interest. Thus from an algorithmic perspective we argue that relevance and fidelity are the basic criteria for useful explanatory examples for human analogical reasoning. In this context we define relevance as visible and recognizable similarity between the input $\mathbf{x}^i$ of the item-of-interest and that of the neighbor $\mathbf{x}^n$. Fidelity we define to mean that the model treats $\mathbf{x}^i$ and $\mathbf{x}^n$ similarly, producing similar outputs $\hat{y}^i$ and $\hat{y}^n$. A good explanatory example retrieval algorithm is one which generally produces examples which are relevant to the item-of-interest and consistent with the model's treatment of that item.

A common formulation of an analogical argument is: given that an object O^1 has properties A, B and C, then an object O^2 with properties A and B probably also has property C. The strength of such an analogy is determined by the pertinence of properties A and B to property C, the degree of similarity between objects O^1 and O^2 on these dimensions, as well as the number and diversity of objects O^1 which are able to serve as premises for the comparison. Fallacies such as bad analogies or slippery slope arguments emerge when these criteria are not fulfilled (*Salmon*, 2012).

In interpretable machine learning, the object of comparison is a relationship between an output $\hat{y}^i$ and an input $\mathbf{x}^i$ –the prediction we are trying to explain and the

input upon which the model produced that prediction. Depending on the exact formulation of our task, we may be trying to estimate the error of the prediction or we may be trying to estimate the true label of the input. Either way, the key properties in this comparison are the input and the output, and if either comparison fails we are at risk of a spurious analogy.

4.1.2 Confidence estimation

Explanatory example retrieval has two faces: human and machine-oriented. We argue above that showing examples to a human user can allow that human to apply analogical reasoning to assessing the reliability of a machine prediction on a given item-of-interest. However, explanatory examples can also be used as part of an automated confidence estimation procedure by deriving indicators of potential model error from aspects of the relationship between an item-of-interests and its neighbors under a given retrieval scheme.

Papernot and McDaniel (2018) explores the latter application by drawing on the idea of conformal prediction (*Shafer and Vovk*, 2007; *Papadopoulos*, 2008) to propose that explanatory examples can be used as a better estimate of model confidence than conventional methods. They use the idea of nonconformity, the extent to which a predicted class disagrees with the true classes of the explanatory examples, as a way to estimate model confidence. In their formulation, the nonconformity α of a predicted class c^i for an input $\mathbf{x}^i$ is the number of retrieved examples with a different class from that predicted:

$$\alpha(\mathbf{x}^i, c^i) \leftarrow \sum_n |c^n \neq c^i| \tag{4.1}$$

The intuition here is that if the model proposes a different output for the item-of-interest than the true labels of many of its nearest neighbors in the chosen retrieval space, then that prediction is more likely to be erroneous than one with greater

agreement.

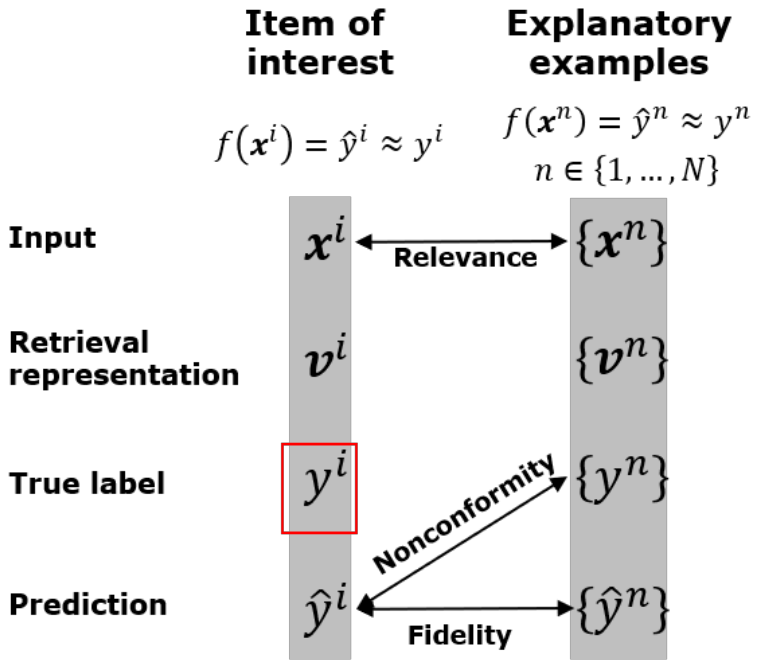

Figure 4.3: Possible comparisons between item-of-interest and explanatory examples. Relevance and fidelity defined above are comparisons of x_i against $\{x_n\}$, and $\hat{y}_i$ against $\{\hat{y_n}\}$ and respectively. Nonconformity as defined by *Papernot and McDaniel* (2018) is a comparison between $\hat{y}_i$ and $\{y_n\}$

However, this is just one comparison that can be made between an item-of-interest and a series of explanatory examples. As Figure 4.3 shows, there are a variety of possible comparisons that can be made between the prediction of the model on the item-of-interest and the predictions of the model on a given set of N explanatory examples. Relevance and fidelity, as defined above, represent comparisons between the inputs $\mathbf{x}$ and predictions $\hat{y}$ for the item-of-interest and the explanatory examples. Nonconformity as defined by *Papernot and McDaniel* (2018) is a comparison between a prediction $\hat{y}^i$ for the item-of-interest and the true labels y^n of the explanatory examples.

We suggest two other potentially useful comparisons: retrieval distance and example error (Figure 4.4).

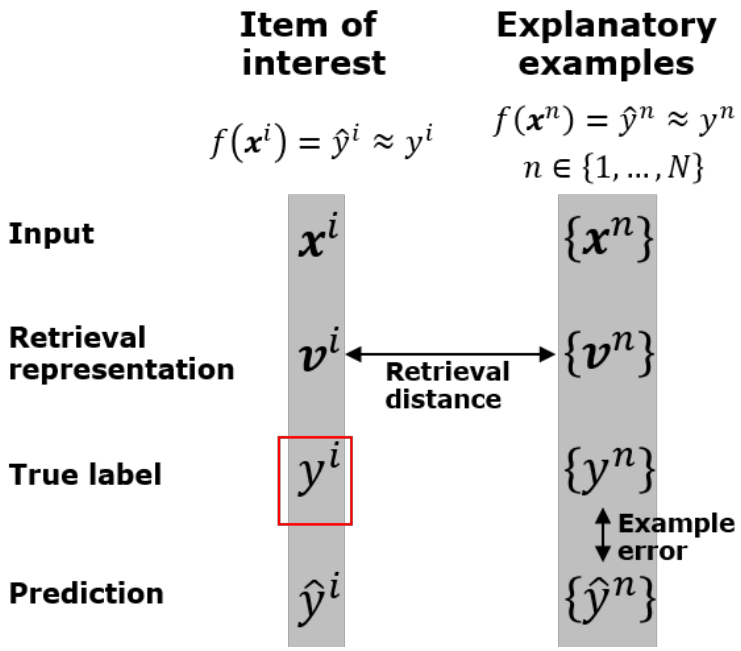

Figure 4.4: Proposed comparisons between item-of-interest and explanatory examples.

We define retrieval distance as the mean distance between retrieved examples and the item-of-interest, by whatever metric is used to retrieve those examples in the first place. The motivation for this comparison is that an item-of-interest for which the only explanatory examples available are relatively far away, may be an item for which the training data lacks support and thus for which the prediction error is likely to be high.

We define example error as the mean prediction error of the retrieved examples. If this error is high, that indicates that the item-of-interest may reside in a region of feature space for which the model was unable to learn a good prediction function, which again may serve as an indicator of increased error likelihood.

Nonconformity, retrieval distance and example error all are potential avenues for gleaning insight about the error of the model on specific items-of-interest. We base part of our empirical evaluation on the extent which the compared retrieval methods produce neighbors where these qualities are predictive of model error on the item-of-

interest.

4.1.3 Contributions

In this chapter we propose to use model attention to achieve the selection of neighbors which are relevant, high-fidelity and predictive of model error, as well as to present of those neighbors in an effective way.

To select explanatory examples for a text classifier's decision about a given item-of-interest, we use a variant of the adversarial attention mechanism proposed in (*Carton et al.*, 2018) to identify the parts of the comment the classifier considered to be indicative of the positive class (in this case, online comment toxicity). We generate vector representations for the whole dataset by using the attention mask for each item as an additional weighting on the embedding centroid method proposed by *Arora et al.* (2017). Finally, we select neighbors for an item-of-interest by using a space-preserving data structure to identify examples from the training set that are close in terms of euclidean distance to this attention-weighted representation.

In an empirical evaluation, we show how the proposed algorithm compares with a selection of baseline methods in the qualities of the retrieved neighborhoods. The result of this evaluation shows that our proposed approach exceeds baseline methods at finding relevant examples while remaining comparable in terms of fidelity and correlates of model error.

The contributions of this chapter are as follows:

- We propose a framework for evaluating the quality of an explanatory example retrieval algorithm.
- We propose a retrieval algorithm that outperforms strong baselines in terms of the proposed framework.

4.2 Methods

The proposed algorithm applies feature attribution to a simple sentence embedding technique based on finding the centroid of the tokens in the given text. It uses a model attention mechanism for feature attribution, and combines it with the centroid method proposed by *Arora et al.* (2017) to find neighbors which have high relevance and fidelity to the model's predictions.

4.2.1 Hypothetical attention

To generate feature importance weights we employ a variant of the model proposed in Chapter II. In the original model (section 2.2), an RNN attention layer produces an attention mask $\mathbf{z}$ which is fed, along with the input $\mathbf{x}$, into an RNN prediction layer to make a prediction $\hat{y}$. The attention and predictive layers are trained in tandem to make accurate predictions with attention masks which, while sparse, are encouraged via the presence of an additional adversarial predictive layer to contain all available predictive signal.

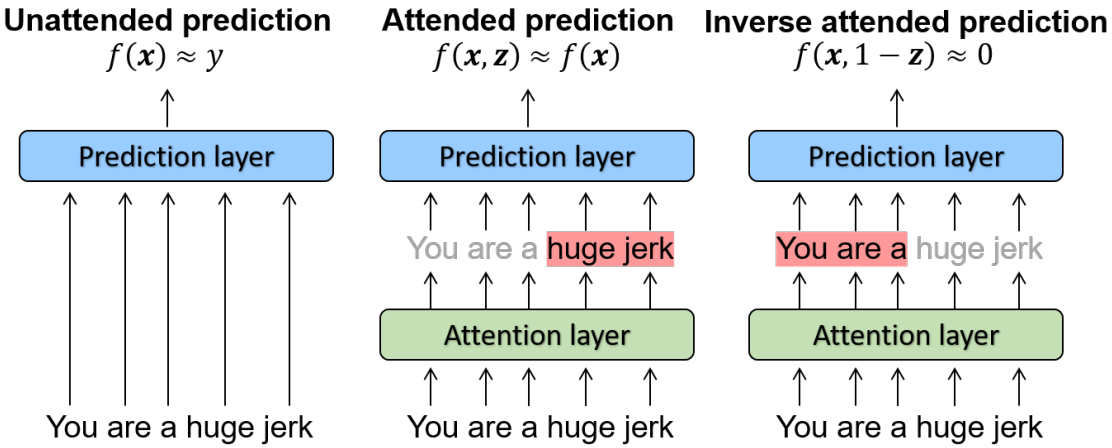

Figure 4.5: Hypothetical attention architecture. The predictive layer is trained to maximize the accuracy of the unattended prediction. The attention layer is trained to push the attended prediction close to the unattended prediction, and the inverse attended prediction close to 0.

In this variant (Figure 4.5), we dispense with the adversarial predictor and don't filter the input to the predictive layer through the attention layer. Hence, the (now solitary) predictive layer f is a standard RNN (an LSTM specifically), training according to a standard squared-loss objective:

$$cost_f(\mathbf{x}, y) = \big[f(\mathbf{x}) - y\big]^2 \tag{4.2}$$

The attention layer is trained in terms of the impact that it *would* have on the primary predictor if it were used to filter input to that layer. Hence, the attention masks produced by this variant model can be thought of as "hypothetical attention". We refer to $f(\mathbf{x})$ as the prediction made by the predictive layer without attention, and $f(\mathbf{x}, \mathbf{z})$ to be the prediction that would have made if the attention mask $\mathbf{z}$ were used as a weighting on the input $\mathbf{x}$. We also refer to the inverse attention prediction $f(\mathbf{x}, \mathbf{1} - \mathbf{z})$, the prediction that the predictive layer would make if the input were masked by the inverse of the attention mask $\mathbf{z}$.

Specifically, the attention layer is trained to produce attention masks which cause the attention-prediction $f(z, x)$ to deviate as little as possible from the non-attention prediction $f(x)$, while simultaneously encouraging the inverse-attention-prediction to be as low as possible. The objective function also includes the sparsity and cohesiveness terms from the original model, leading to the following objective function:

$$cost_g(z, x, y) = \tag{3}$$

$$\big[f(x, z) - f(x)\big]^2 \tag{3.1}$$

$$+\lambda_1 ||z|| \tag{3.2}$$

$$+\lambda_1 \lambda_2 \sum_t |z_t - z_{t-1}| \tag{3.3}$$

$$+\lambda_3 \big[f(x, 1 - z) - f(x, 0)\big]^2 \tag{3.4}$$

Term 3.1 encourages the attention mask to alter the model's prediction as little as possible. Terms 3.2 and 3.3 encourage the model to produce sparse but "chunky" attention masks favoring long strings of similar weights.

Term 3.4 is where this variant differs from the original architecture. Instead of encouraging the attention layer to disallow an independent adversarial predictor from finding any predictive signal in the inverse-masked input, it instead encourages that layer to disallow the primary predictor from doing this, while the primary predictor does not learn adversarially with respect to this objective in the way that the adversarial predictor does in the original model.

Ultimately this accomplishes something very similar to the algorithm of Chapter II. In the toxicity detection task it encourages the attention mask to retain the toxic content which drives the model's prediction while also encouraging it not to leave any toxic content out of the mask. This algorithm ends up being similar to that described in (*Li et al.*, 2016), though the objective is slightly different and it does not employ hard attention.

Where this variant differs from the original algorithm is there is no adversarial predictor attempting to detect trace evidence of toxicity in the residual of the existing attention mask. What this means is that the attention mask catches the content that drives the model's predictions, but not necessarily all the content that *could* drive those predictions. This translates to somewhat lower recall of toxic content. Dispensing with the adversarial predictor also obviates the need for the confusion step described in section 2.2.2.

However, the advantage of this variant approach is that it produces a predictive layer which is indistinguishable from an ordinary LSTM trained with an ordinary objective function. This allows us to make a direct comparison between our proposed explanatory example retrieval method and baseline methods based on the extraction of latent vector representations from the LSTM model.

4.2.2 Attention-weighted word centroids

Explanatory example retrieval seeks to identify items from the labeled dataset which explain or otherwise helpfully contextualize the model's prediction on a given item-of-interest (IOI). Doing so involves two steps: 1) generating a useful representation of items; 2) choosing a distance metric and retrieval method.

A common explanatory representation for neural networks is the final layer of the model–that is, the final vector produced by the model before it is transformed via sigmoid or softmax into an output prediction (*Caruana et al.*, 1999; *Wallace et al.*, 2018; *Papernot and McDaniel*, 2018). The rationale for this approach is that by the time inputs have reached this point in the model, they should be linearly separable, residing in a latent vector space in which inputs of different classes are located on opposite sides of the unit hyperplane. Points that are in close in this space, then, should be points with both a similar predicted class and similar values on what the algorithm determined to be the most useful linear factors of the desired prediction task.

However, when applied to RNN models for text classification, this method has a tendency to retrieve examples which are incoherent–unrecognizable to a human as being related to the item-of-interest, a result noted in *Wallace et al.* (2018) and which we demonstrate in the empirical evaluation.

Our alternative approach is simple. We generate explanatory examples by applying the weights produced by the attention layer described above to the embedding centroid method described in *Arora et al.* (2017). This method embeds a word sequence by finding the centroid of the individual word embeddings, weighted by the corpus frequency of each word. Finally, the method computes the principal component of the full set of embeddings over the whole vocabulary and then subtracts the first component from all sentence embeddings.

Incorporating attention weights into *Arora et al.* (2017), the explanatory represen-

tation of an input item $\mathbf{x}$ of length l composed of words $\{w_0, ..., w_l\}$ with embedding vectors $\{v_0, ..., v_l\}$ of dimensionality d, for which the attention layer $g(\mathbf{x})$ has produced an attention mask $\mathbf{z} = \{z_0, ..., z_l\}$ is calculated in two steps. First, the unadjusted weighted embedding centroid $\mathbf{v}_x^{'}$ is produced as follows:

$$\mathbf{v}_x^{'} \leftarrow \frac{1}{l} \sum_{l}^{i} \frac{a}{a + p(w_i)} z_i v_i \tag{4.4}$$

Here, a is a smoothing constant set at 0.001, and $p(w_i)$ the frequency of word i, which we calculate from the training set.

In the second step, the unadjusted centroids $\{v_{x0}^{'}, ..., v_{xm}^{'}\}$ of all m training set items are concatenated together as columns of a large matrix V of dimension $l \times d$. The first principal component of this matrix is calculated as $\mathbf{u}$, and is subtracted from the unadjusted centroid to form the final vector representation of the input x:

$$\mathbf{v}_x \leftarrow \mathbf{v}_x^{'} - \mathbf{u}\mathbf{u}^T\mathbf{v}_x^{'} \tag{4.5}$$

The effect of this procedure is to represent the input sequence $\mathbf{x}$ as the centroid of only those words in the sequence that were impactful to the model's prediction. The embeddings that are produced have fidelity to the model in the sense that they are based on the model's feature importance weights, but are also comprehensible to a human user because they are based on word similarity between the texts.

We include the unaltered centroid method as a baseline method for comparison.

4.3 Empirical Evaluation

We perform an empirical evaluation of the proposed attention-weighted centroid retrieval method to show how it performs in terms of the criteria described in section 4.1. In terms of the taxonomy proposed by *Doshi-Velez and Kim* (2017), this is a functionally-grounded evaluation extending from the principle that a good ex-

planatory example should be of high relevance and fidelity and should be useful in predicting model error.

We compare our proposed attention-weighted word embedding centroid method with a number of baselines, including:

- Random: Neighbors are sampled randomly from the dataset and assigned a random distance between 0 and 1.
- Output: Nearest neighbors in terms of model prediction.
- Pre-output layer: Nearest neighbors in euclidean distance between model pre-output layer (e.g. *Wallace et al.* (2018)).
- Corpus-frequency weighted wording embedding centroids (*Arora et al.*, 2017).

All retrieval methods use the same model, meaning that all model predictions and attention masks are identical across retrieval methods. The primary baseline is the pre-output layer method, as this is a commonly suggested technique for identifying explanatory neighbors for neural nets (e.g. *Wallace et al.* (2018); *Caruana et al.* (1999).

The model is trained on the toxicity dimension of the dataset described in *Wulczyn et al.* (2017), which consists of roughly 100,000 training instances, 30,000 development instances and 30,000 test instances. Our evaluations are performed on the test set using neighbors drawn from the training set.

4.3.1 Relevance and fidelity

We evaluate each algorithm in terms of the mean relevance of retrieved neighbors as well as the fidelity of the model's prediction on those neighbors with its prediction on the item-of-interest (Table 4.1). All metrics are based on the closest 3 neighbors generated by each method.

Relevance is operationalized as mean jaccard similarity between neighbor and item-of-interest text. We report both an unweighted jaccard similarity which reflects

the overall text similarity, and an attention-conscious jaccard similarity that only considers tokens whose attention weight was above a threshold of 0.25. This latter value measures the retrieval method's success in finding examples with similar attention masks to that of the item-of-interest.

We also report the mean structural difference between item-of-interest and neighbor texts, defined as the mean absolute difference between text lengths for both text and attention-thresholded text.

We report fidelity by treating the mean neighbor prediction $\{\hat{y}^n\}$ as an estimate of the item-of-interest prediction $\hat{y}^i$, and then reporting the mean absolute error and accuracy of this prediction (the latter with respect to binarized versions of $\{\hat{y}^n\}$ and $\hat{y}^i$). A perfectly faithful neighborhood with $\{\hat{y}^n\}$ identical to $\hat{y}^i$ will thus have zero error and perfect accuracy on this metric.

	Semantic relevance (Jaccard similarity)		Structural relevance (Text length difference)		Fidelity	
	Text	Attributed text	Text	Attributed text	MAE	Acc.
Random	0.072	0.001	94.455	6.554	0.172	0.896
Output	0.083	0.013	92.081	4.983	**0**	**1**
Pre-output layer	**0.163**	0.054	**42.007**	**2.823**	**0.017**	**0.992**
Attention centroid	0.141	**0.222**	72.206	5.272	0.046	0.972
Centroid	0.172	0.02	216.267	15.352	0.082	0.954

Table 4.1: Comparison of relevance and fidelity metrics across different retrieval algorithms.

Table 4.1 summarizes the results. The results show that while the pre-output layer method is the most successful method at finding neighbors of high general text similarity, it fails to find neighbors of high *attributed* text similarity. However, this retrieval method is successful at finding neighbors of high attributed *structural* similarity, tending to find neighbors with attention masks of similar length to that of the item-of-interest. What this suggests is that the internal representation learned by the LSTM is more structural than semantic, remembering roughly how many toxic

words were encountered but not what they were. Figure 4.2.2 is a demonstration of this phenomenon: the attention mask of that neighbor has the same length but consists of a different word than that of the item-of-interest.

By contrast, our proposed attention-weighted centroid method produces neighbors of slightly less general text similarity but much higher attributed text similarity, succeeding much better at finding neighbors which are recognizably relevant to a given item-of-interest. Meanwhile, while the pre-output layer method retrieves neighbors of almost perfect model prediction fidelity to the item-of-interest, our proposed method retrieves neighbors of only slightly less fidelity. Understandably, perfect fidelity is achieved by finding neighbors based solely on the model's output, but this trivial method finds neighbors which are not otherwise particularly similar to the item-of-interest.

Hence, while our algorithm is slightly more likely to produce low-fidelity explanatory examples, it is much more likely to succeed at producing relevant examples, and thus to fulfill the criteria for analogical validity outlined in the introduction.

4.3.2 Predicting model classification error

We evaluate each retrieval method in terms of its ability to produce explanatory examples which can be aggregated into neighborhood metrics that are predictive of model classification error, namely neighbor nonconformity, neighbor retrieval distance and neighbor error. We define these metrics below:

Nonconformity measures the extent to which the predicted class of the item-of-interest agrees with the true classes of the retrieved neighbors:

$$nonconformity(\mathbf{x}^i) \leftarrow \frac{1}{n} \sum_n |c^n - c^i| \tag{4.6}$$

Neighbor retrieval distance measures the mean euclidean distance between the

item-of-interest and its neighbors, in terms of the retrieval representation v of each item:

$$distance(\mathbf{x}^i) \leftarrow \frac{1}{n}\sum_n (v^i - v^n)^2 \tag{4.7}$$

Neighbor error measures the mean absolute prediction error achieved by the model on the retrieved neighbors for the item-of-interest:

$$error(\mathbf{x}^i) \leftarrow \frac{1}{n}\sum_n |y^n - \hat{y}^n| \tag{4.8}$$

For each retrieval method/neighborhood metric pair, we perform the following procedure:

1. Retrieve the top three neighbors for every item in the development and test sets using the given retrieval method

2. Train a logistic regression model on the development set, predicting classification error using the given neighborhood metric as the sole input feature.

3. Evaluate trained model on the test set using cross-entropy of predicted error probability

We follow *Guo et al.* (2017) in using cross-entropy as an evaluation metric for this task. For a given item $\mathbf{x}^i$, our model produces a class probability $\hat{y}^i$ which is optimized to be as close as possible to the true class probability y^i. We can treat this as a classification problem by binarizing $\hat{y}^i$ and y^i to class labels $\hat{c}^i$ and c^i respectively and then considering the classification error of this prediction $e_c^i = 0$ if $\hat{c}^i = c^i$ else 0. If we consider a confidence estimation to be a prediction about the probability of the classification error $p_e^i = p(e_c^i = 1)$, then we define the cross-entropy loss l_e^i as follows:

$$l_e^i \leftarrow e_c^i log(p_e^i) - (1 - e_c^i) log(1 - p_e^i) \tag{4.9}$$

This metric measures the ability of the method to produce error probabilities which are consistent with the empirical error distribution. A useful upper bound on error for this metric is the trivial approach of using the development set accuracy of 95.5% as a confidence estimate for every item in the test set. Doing this results in a cross-entropy confidence error of 0.181 on the test set.

The toxicity dataset of *Wulczyn et al.* (2017) differs from many classification datasets in providing class probabilities as training data rather than class labels alone. This allows us to train an unusually well-calibrated classifier where the distance of the true and predicted class probability from the decision boundary of 0.5 can serve as an estimate of model error. We find that using these predicted class probabilities as a confidence estimate results in a cross-entropy loss of 0.124, while using the true class probabilities results in a cross entropy loss of .11. These values serve as soft lower bounds on the cross-entropy error of the proposed neighbor-based methods in the sense that it would be very difficult for improve on them.

	Non-conformity	Retrieval distance	Neighbor error	All methods
Random	0.177	0.181	0.181	0.177
Output	0.159	0.181	**0.159**	0.159
Pre-output layer	0.156	0.181	0.162	0.155
Attention centroid	**0.15**	**0.176**	0.168	**0.149**
Centroid	0.165	0.181	0.178	0.164

Table 4.2: Comparison of model confidence estimation cross-entropy (with respect to true classification error) across different retrieval and estimation methods. 0.181 represents a trivial result.

Table 4.2 summarize the result of using each metric as a predictor of binary model classification error, as well as the result of combining all three metrics into one predictive model. What the results show is that neighbor nonconformity is generally the best-performing indicator of model error, while our proposed retrieval method produces neighbors which are more predictive of model error for two of three confidence estimation methods, as well as when all three methods are combined into one

confidence estimation model.

Our proposed method is the only one tested in which neighbor distance is a meaningful predictor of model error, but it does not add much predictive utility to nonconformity when incorporated into a combined model of classifier error.

The nontrivial result of nonconformity for the random neighbor/random distance retrieval method can be explained by the non-independent class and error distribution of the *Wulczyn et al.* (2017) dataset. The dataset is 89% negative (nontoxic), and the model achieves a binary accuracy of 98.5% on these negative examples but only 71.9% on the positive 11% of the dataset. So, when a random three neighbors are chosen for a given $\mathbf{x}^i$, these neighbors will mostly be of the negative class and therefore of low nonconformity to a negative $\mathbf{x}^i$ (on which the model is liable to be accurate) and of high nonconformity to a positive $\mathbf{x}^i$ (on which the model is liable to be inaccurate). Hence, the nonconformity of random neighbors is predictive of the predicted class of the item, which itself is predictive of model error.

Taken as a whole, the empirical evaluation demonstrates that our proposed method of retrieving neighbors based solely on attributed content results in explanatory examples which are significantly more visually relevant, only slightly less faithful to the model's treatment of the item-of-interest, and slightly more predictive of model unreliability than comparable baseline methods.

4.4 Discussion

This chapter describes an automated evaluation which compares the success of several explanatory example retrieval algorithms in selecting neighbors that are analogically valid and faithful to model behavior, as well as their potential to reveal insights about the predictive error of the model. Our approach uses feature attribution to assign importance weights to every token of a given input, and then retrieves examples by finding items with similar important tokens.

The results of this evaluation show that our proposed algorithm is much more able to retrieve neighbors with similar attribution masks than competing algorithms, while not suffering much in terms of fidelity to model behavior on those neighbors (Table 4.1). At the same time, our proposed method performs better than baseline methods in producing useful predictors of model error (Table 4.2)

One interesting result of this evaluation is how unaligned the pre-output layer of the LSTM is with feature attribution performed against its predictions–nearest neighbors retrieved using this final representation within the model generally do not have similar attributed features to their items-of-interest. However, this retrieval method *does* succeed in finding texts of similar lengths, and with attributed sections of similar sizes. What this implies is that much of what is saved into the output layer of an LSTM model is of a structural rather than a semantic nature, recording only that the predictive content was encountered rather than what that content was.

Figure 4.6 demonstrates this tendency in action. The pre-output layer finds a neighbor with a similar one-word attention mask, but it is a different (and semantically dissimilar) word from that used in the comment-of-interest.

Our attention-based proposed algorithm maintains a substantial degree of fidelity to the model behavior while producing neighbors that align better in terms of attributed content. However, our method does remain intrinsically decoupled from the model because it is based on the input representation and thus does not directly reflect the model's decision in the way that the pre-output layer does.

One solution that could combine the best of both worlds–the relevance of our method with the model fidelity of the pre-output layer–would be to place a generative objective on the pre-output layer that would encourage it to be able to reconstruct the (attributed) input text. Thus, the model would be optimized not only to be able to make accurate predictions from the pre-output layer, but also to recover the input tokens, with a high weight attached to tokens that were of high impact in the

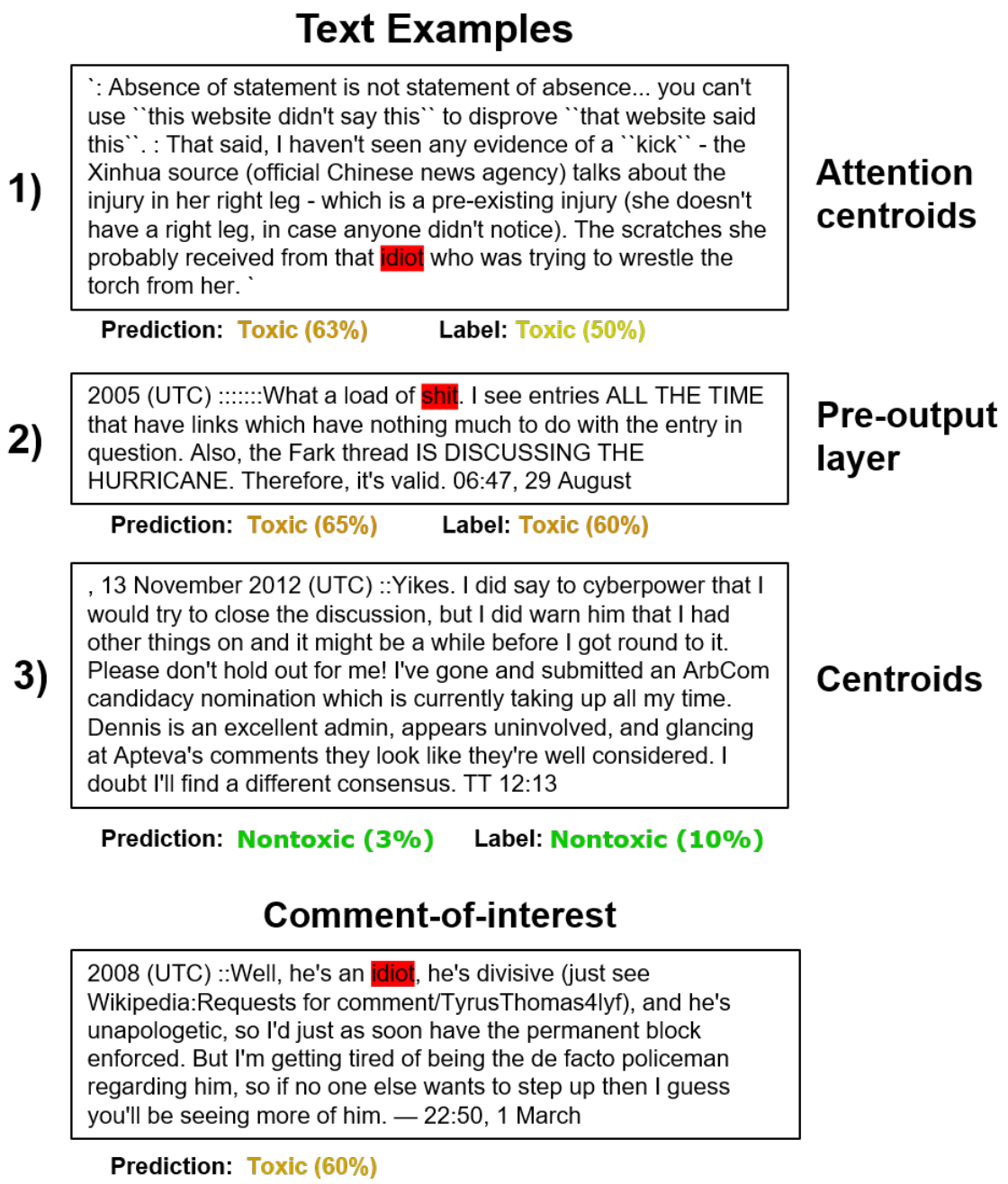

Figure 4.6: Closest neighbor for each algorithm tested in the user study: 1) attention centroids; 2) pre-output layer; 3) centroids.

prediction and a low weight to those that were less impactful.

A sequence-to-sequence architecture (*Sutskever et al.*, 2014) would be a natural fit for this idea. It is possible, however, that the requirement of reconstructing the

original sequence in order would be unnecessary and overly strict. In this case, it might work better to place a non-sequence-based decoder on the pre-output layer, such as the decoder half of a skip-gram model (*Mikolov et al.*, 2013), which would optimize simply to recover the correct tokens without worrying about the sequence order. In either architecture it would be relatively straightforward to fold in the feature attribution as a weighting on which tokens needed to be recovered and which could be ignored.

4.4.1 Hybrid robustness

Like Chapter II, our empirical evaluation in this chapter proceeds from an assumption about how human subjects are likely to use example-based methods in order to improve their estimates about the consensus toxicity of social media comments. We assume that subjects will perform analogical "this is like that" reasoning based on the similarity between the texts and the model outputs for those texts. Thus we assume that a proper example is both visibly similar to the item-of-interest (relevance) as well as treated similarly by the model (fidelity) so that the human subject does draw any incorrect inferences from the the model's output on that item.

Like our assumption about the importance of high-recall attribution masks, this conceit is reasonably well-supported by the dictionary definition of analogical reasoning, but it requires a user study to see if it holds up in practice. We perform such a study in Chapter V.

CHAPTER V

User Study 2: Effect of Example-based Explanations

5.1 Introduction

The empirical evaluation we describe in chapter IV shows that our proposed algorithm retrieves neighbors with more similar attributed content than competing baselines and comparable useful numerical qualities such as prediction fidelity and correlation with model error.

However, as is the case with any approach to interpretable machine learning, performance on proxy measures is only one part of the story. While such an evaluation provides indicators of the potential for such an algorithm to improve human performance, a user study is needed to assess whether it actually achieves this improvement. We describe such a study in this chapter.

The second part of our evaluation consists a user study that compares the relative desirability and utility of three of the nearest neighbor selection algorithms discussed in the previous chapter:

1. **Pre-output layer**: The final parameter-weighted vector that is passed into the

[0]This chapter and the previous one consist of content published in
Carton, S., Mei, Q. and Resnick, P. (2020). Model Attention for Example-Based Explanations of Text Classifiers, *In submission*.

output sigmoid of the LSTM.

2. **Attention centroid**: Our proposed attention-weighted embedding centroid algorithm, described in subsection 4.2.2

3. **Centroid**: The unaltered word embedding centroid algorithm proposed by *Arora et al.* (2017).

5.2 Experiment design

Our user study compared the relative desirability and utility of the three algorithms listed above. Subjects in our study participated in one of three related sub-studies:

1. **Ground truth collection and baseline prediction**: workers provide their own opinion about the toxicity of items, as well as predicting the mean perceived toxicity of each item without any algorithmic assistance. This is primarily a label collection task, but also assessed the baseline unassisted performance of human workers.

2. **Neighbor preference**: subjects choose between neighbors suggested by the three algorithms. This is a within-subject experiment testing which retrieval algorithm subjects prefer when given a choice of neighbors.

3. **Neighbor prediction**: subjects make predictions about toxicity assisted by neighbors drawn from a single algorithm. This is a between-subject experiment of whether neighbors helped subjects make better predictions.

In discussing the latter two experiments, we draw a distinction between the comments-of-interest that subjects were asked to evaluate predictions about and the neighbor comments that were used to explain those predictions. In the study we used

the term "main comment" to refer to the comments-of-interest, so we use these terms interchangeably throughout the rest of this chapter. We also follow Chapter IV in using the terms neighbor and example interchangeably.

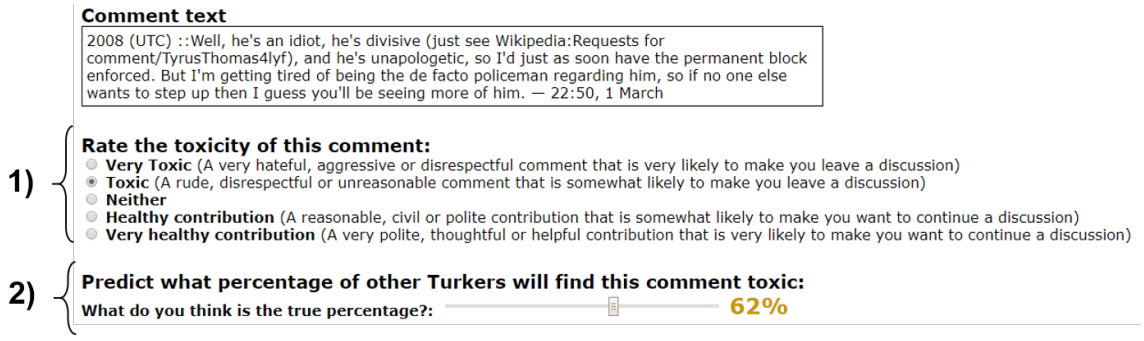

Figure 5.1: Ground truth and baseline prediction task. Subjects are asked to 1) provide a subjective label; 2) make a prediction about their own population.

5.2.1 Ground truth collection and baseline prediction

In the ground truth and baseline prediction task (Figure 5.1), subjects reviewed a series of comments-of-interest drawn from the *Wulczyn et al.* (2017) dataset. For each comment, they gave their personal judgment about whether the comment was toxic or not according to the same questionnaire used in *Wulczyn et al.* (2017). They were also asked to use a range slider to predict what percentage of other workers were liable to find each comment toxic or very toxic.

The purpose of this task was to collect low-variance ground truth toxicity scores for the selected comments, as well as an understanding of the baseline accuracy of workers in predicting the toxicity perceptions of their own cohort.

5.2.2 Neighbor preference

The preference experiment was a within-subject experiment designed to behaviorally evaluate which neighbor retrieval algorithm subjects preferred when asked to

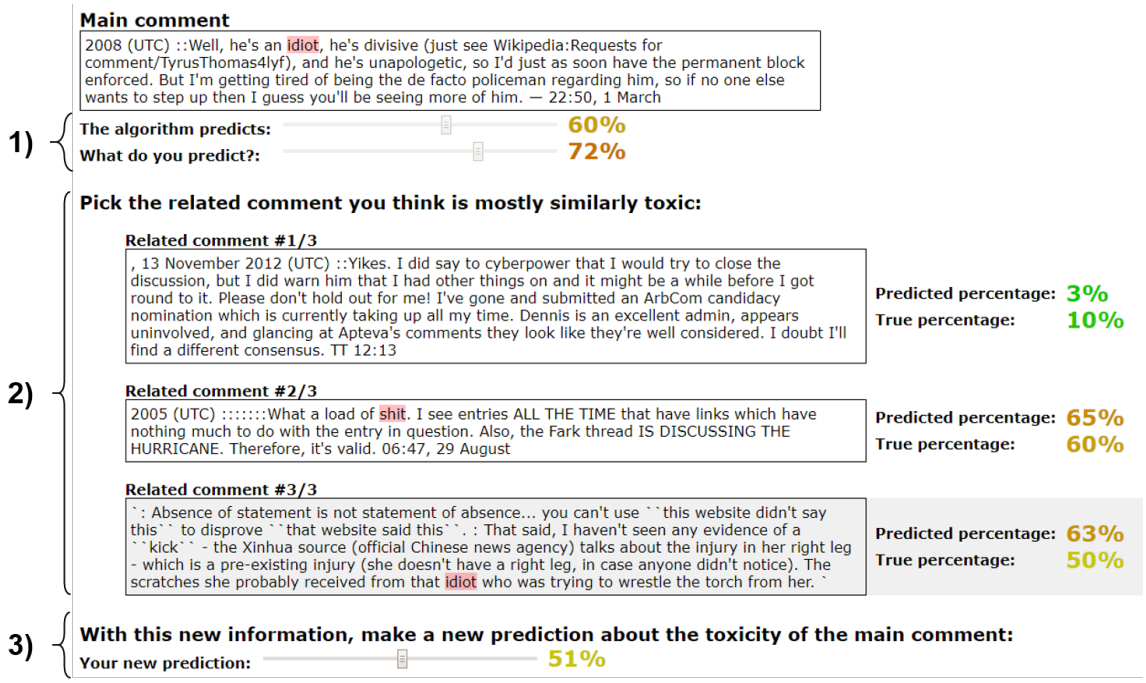

Figure 5.2: Algorithm preference task. Users are asked to 1) make an initial prediction; 2) choose an evidence comment; 3) make a final prediction.

use those neighbors to help perform a prediction task.

In this task (Figure 5.2), subjects were presented with a series of comments as well as a model prediction and a feature attribution mask over the text in the form of highlighting of putatively toxic content. The prediction and highlighting were generated by the variant of the Chapter II model described in subsection 4.2.1. Subjects were asked to provide three responses for each comment:

For the first input (Figure 5.2.1), subjects were presented with the model's prediction and feature attribution about the toxicity of the comment and asked to make their own prediction about what percentage of other workers are liable to find that comment toxic.

For the second input (Figure 5.2.2), subjects were presented with three explanatory neighbors–the nearest neighbor in the training set found by each of the three methods. Each neighbor was highlighted as appropriate by the attention mechanism

described above, but otherwise presented alone.

Subjects were asked to select the single neighbor they felt would provide the strongest evidence about the true toxicity of the main comment. The instructions elaborated on this goal by suggesting the subject choose the most "similarly toxic" comment to the main comment. The reason we provided this particular nudge was to encourage subjects to try to select the neighbor that was the most similar in terms of the target outcome (toxicity) to each item-of-interest. We found in pilot testing that subjects felt confused about what could constitute "evidence" in the context of the presented decision problem.

For the third input (Figure 5.2.3), subjects were presented with the predicted and true toxicity for each neighbor, and asked to adjust their initial prediction based on any insight they may have gotten from the displayed neighbors (with a visual reminder showing them which neighbor they selected in the previous step).

The purpose of this experiment was to gauge neighbor preference by examining which algorithm users choose the selected neighbor from (noting that subjects were not aware that neighbors were being produced by multiple algorithms). The purpose of the two prediction steps was to incentivize subjects to select a neighbor they felt would be predictively useful. For that reason we did not analyze nor report the prediction results of this experiment–only the neighbor selection results.

5.2.3 Prediction with neighbors

Finally, the prediction experiment was a between-subject experiment designed to gauge whether subjects were able to benefit from the presence of explanatory examples generated by any of the three compared algorithms. This experiment contained three conditions, one for each neighbor retrieval algorithm. Any given subject was only shown neighbors generated by one algorithm, as opposed to the preference task where subjects were shown one neighbor from each.

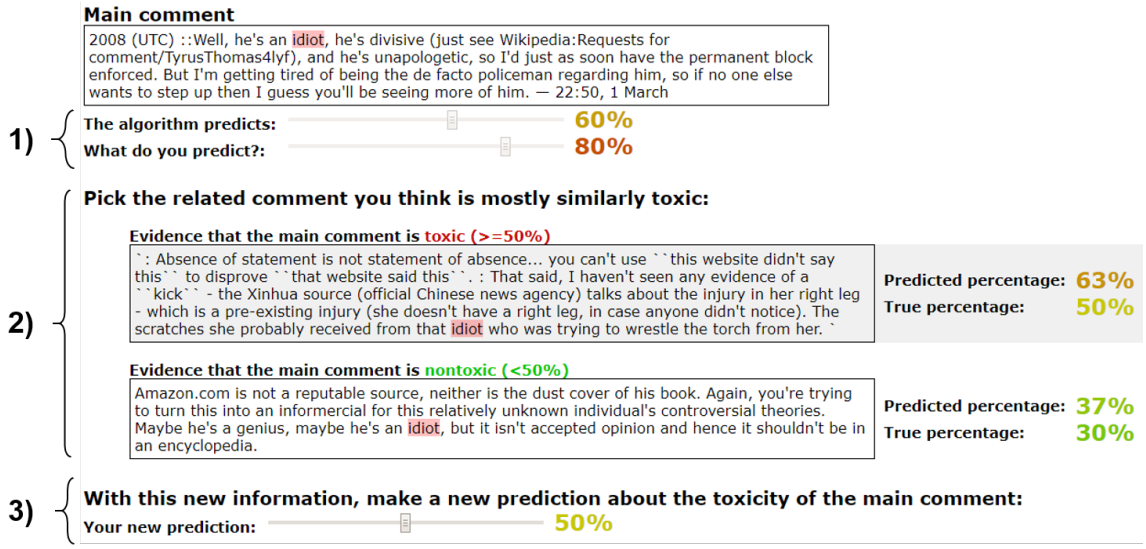

Figure 5.3: Prediction-with-evidence task. Users are asked to 1) make an initial prediction; 2) choose an evidence comment; 3) make a final prediction.

The prediction experiment (Figure 5.3) asked subjects to perform a similar three-stage task for each comment as the preference experiment. Subjects were asked to 1) make an initial prediction; 2) Select a related comment; 3) make a final prediction;

Like the preference condition, subjects were asked in part 2 of their labeling task for each comment (Figure 5.3.2) to select the comment they believed represented the strongest evidence about the true toxicity of the main comment.

However, instead of showing subjects the closest neighbor generated by each algorithm, subjects in the prediction condition were shown two comments retrieved by the same algorithm: the closest neighbor of each true class. That is, they were shown the nearest neighbor whose labeled toxicity was above 50% (making it majority toxic) and the nearest neighbor whose labeled toxicity was below 50% (making it majority nontoxic). Each neighbor comment was displayed with feature highlighting as well as an indication of which class it belonged to (Figure 5.3.2).

The reason for selecting one neighbor from each class is that it was found in pilot testing that simply presenting the N nearest neighbors from a single algorithm would

often result in a selection of very similar neighbors, none of which was clearly stronger than the others. Part of the purpose of this condition was to gauge whether our retrieval algorithm would allow users to make a better choice about which neighbor to regard as the strongest analogy for the item-of-interest. Selecting the closest neighbor of each class allowed more diversity in this choice, as well as a clear winner when compared to the true class of the item.

In addition to measuring the accuracy of the chosen neighbor, we measured the change in error between the subject's initial and final prediction. That is, the extent to which the presentation of the explanatory examples nudged user predictions toward or away from the true toxicity of each comment, as collected in the ground truth/baseline prediction task described above.

5.2.4 Subjects

As stated above, the experiment consists of three sub-studies: the ground truth/baseline prediction task, and the neighbor preference and neighbor prediction experiments. The neighbor prediction experiment was divided into three conditions corresponding to the three retrieval algorithms compared in this study.

50 subjects were recruited for the ground truth task, and 50 for the preference experiment. 30 subjects each were recruited for each of the three prediction conditions. The entire study was repeated across two distinct comment-of-interest sets, leading to a total subject count of 380.

This subject count was chosen as a result of a simulated power analysis which showed it would be able to detect clinically significant effect sizes (which we define as 0.1 for the preference task and 0.05 for the prediction task) in the outcome measures (discussed in the result section below) with the standard minimum probability of 80%. We used a pilot study to estimate outcome variances for this analysis.

Subjects were recruited via, and participated in the study on, the Mechanical

Turk crowdworking platform. Subjects were required to be based in the US and to have completed more than 999 Mechanical Turk human intelligence tasks (HITs) with more than a 98% acceptance rate.

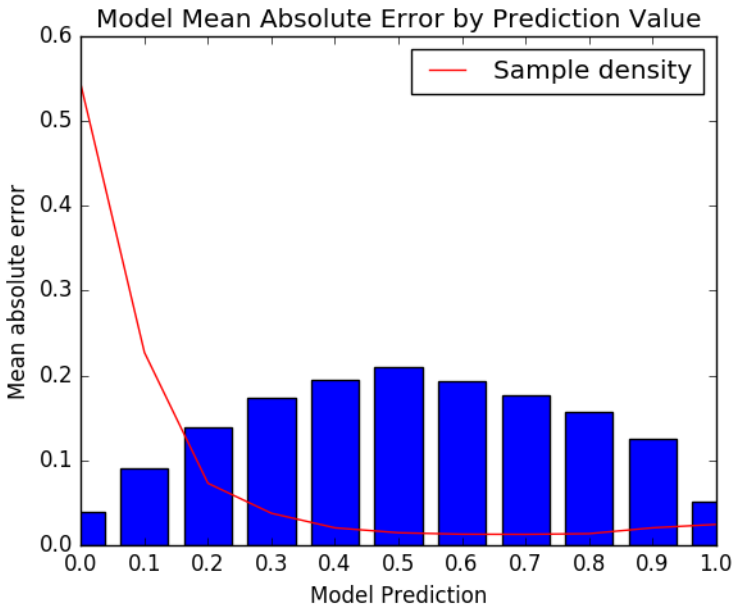

Figure 5.4: Mean absolute error of model across model prediction values for the test set. The red line indicates sample density: the dataset is very unbalanced.

Subjects were compensated via a combination of base pay, attention checks and performance bonuses. The base pay ($0.50 for the ground truth task and $1.00 for the two experiments) was augmented by a flat $0.25 bonus per caught attention check (of which there were 4 in every condition). Subjects in were also given a bonus for accuracy relative to the average accuracy of their condition in predicting the true toxicity of the comments they labeled. This bonus was calculated by setting a maximum possible accuracy bonus for perfect accuracy at $1.00 for the ground truth task and $1.50 for the preference and prediction experiments, and setting average performance to receive half these maximum bonuses. Subjects were then awarded performance bonuses on a linear scale between these two points.

Any subject who missed more than one attention check was discarded and is not

	Selection proportion	Subject prediction error		
		Initial	Final	Change
Pre-output layer	0.297	0.198	0.229	+0.031**
Attention centroids	0.443***	0.2	0.212	+0.013
Centroids	0.26	0.195	0.204	+0.009

Table 5.1: Comparison of subject prediction performance across retrieval algorithms

reflected in the results described below. Only 1 HIT from the preference condition and 3 HITs from the prediction condition had to be discarded in this way.

5.2.5 Comment sampling

Each subject reviewed 13 comments sampled from the *Wulczyn et al.* (2017) dataset. These comments were sampled from the subset of the test set for which the model predicted a toxicity level between 0.4 and 0.6. As figure 5.4 shows, model error is disproportionately high on this segment of the data. This scheme also aligns with other work such as *Nobata et al.* (2016), which suggests hybrid moderation systems that selectively seek human input for borderline instances specifically.

5.3 Results

The outcomes we measure include: neighbor preference; chosen neighbor error; and error difference between initial and final prediction. Table 5.1 summarizes the results of both the preference and prediction tasks.

In the preference task, subjects preferred neighbors retrieved by our proposed algorithm by a significant margin ($p < 0.005$)[1] (Figure 5.5). This outcome is in line with the relevance outcome described in section 4.3.1 above, which demonstrates that our algorithm is much more likely to retrieve neighbors with a similar attention mask

[1]Pairwise independent t-tests between all conditions with Benjamini-Hochberg correction incorporating all outcomes with a target FDR of 0.05.

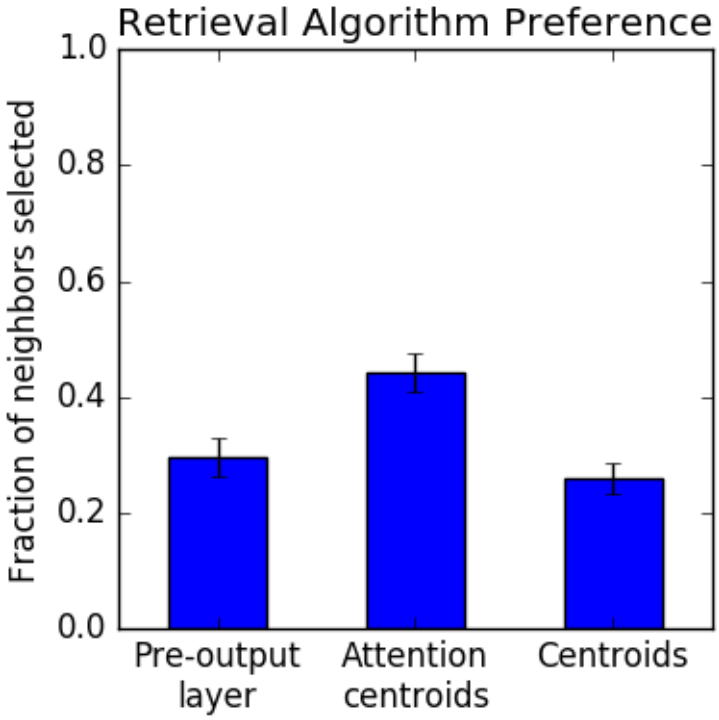

Figure 5.5: Fraction of comments for which each retrieval algorithm was selected in preference task.

to the item-of-interest than other baselines.

In the prediction task, however, in none of the three conditions did the presence of neighbors improve human performance below its mean absolute error of 0.214. Rather, the presence of neighbors actually caused mean absolute error to increase on average (Figure 5.6). This increase in error ranged from marginal ($p < 0.1$) in the case of the centroids and attention-weighted centroids methods to statistically significant ($p < 0.005$) in the case of the pre-output layer method [2].

Examining which neighbors subjects chose in the prediction task helps explain this result (Table 5.2). Subjects in this task were presented with two neighbor comments, one with a true toxicity score above the classification threshold of 0.5 and one below (Figure 5.3). Our hypothesis was that our proposed retrieval method would allow

[2]Related two-tailed t-test between initial and final prediction error for each condition with 3-fold Bonferroni correction.

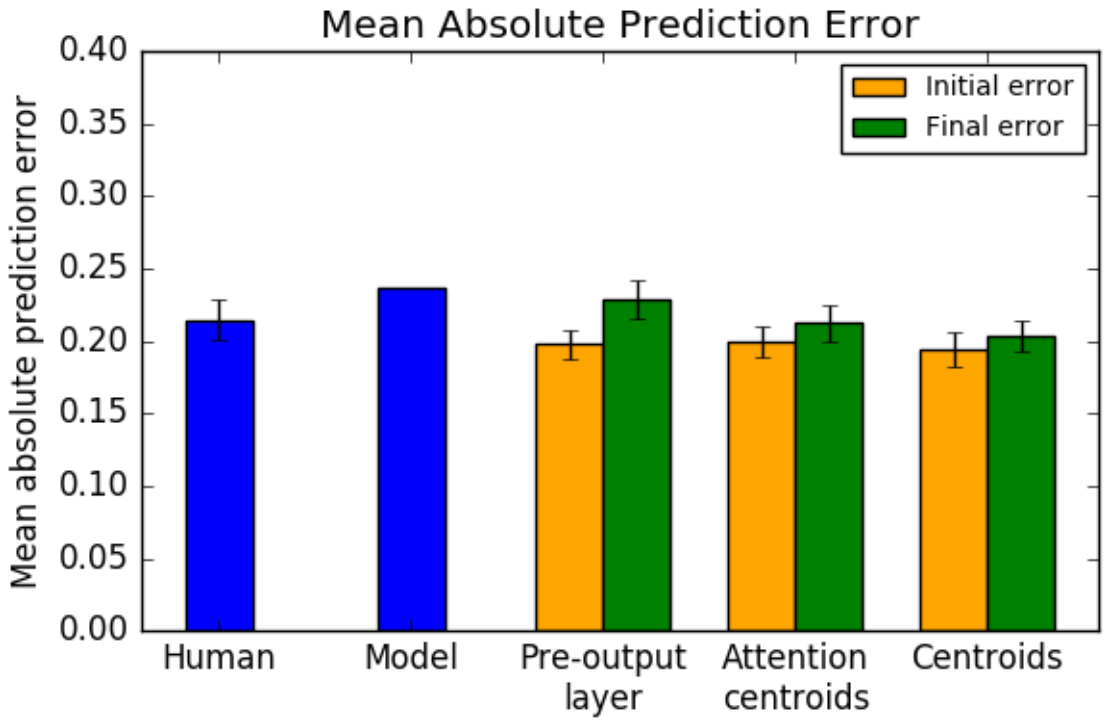

Figure 5.6: Mean absolute initial and final user error across condition, compared to model error and baseline user error.

subjects to choose neighbor comments with a true toxicity closer to that of the item-of-interest.

As Figure 5.7 shows, subjects were able to "beat the mean" in all conditions by selecting neighbors whose true toxicity error was less than the mean error of the presented neighbors. They were able to beat it by the widest margin in the unweighted centroid condition. This result may possibly be because in this condition there tended to be a wider margin of error between the two presented neighbors, thus making for a more clear and meaningful choice between the two neighbors.

However, in none of the three conditions were subjects able to choose neighbors with a mean true toxicity error lower than the baseline human predictive performance. Therefore, even if their final guess was simply a reflection of the true toxicity of their chosen neighbor, subjects were not able to reliably choose neighbors with lower

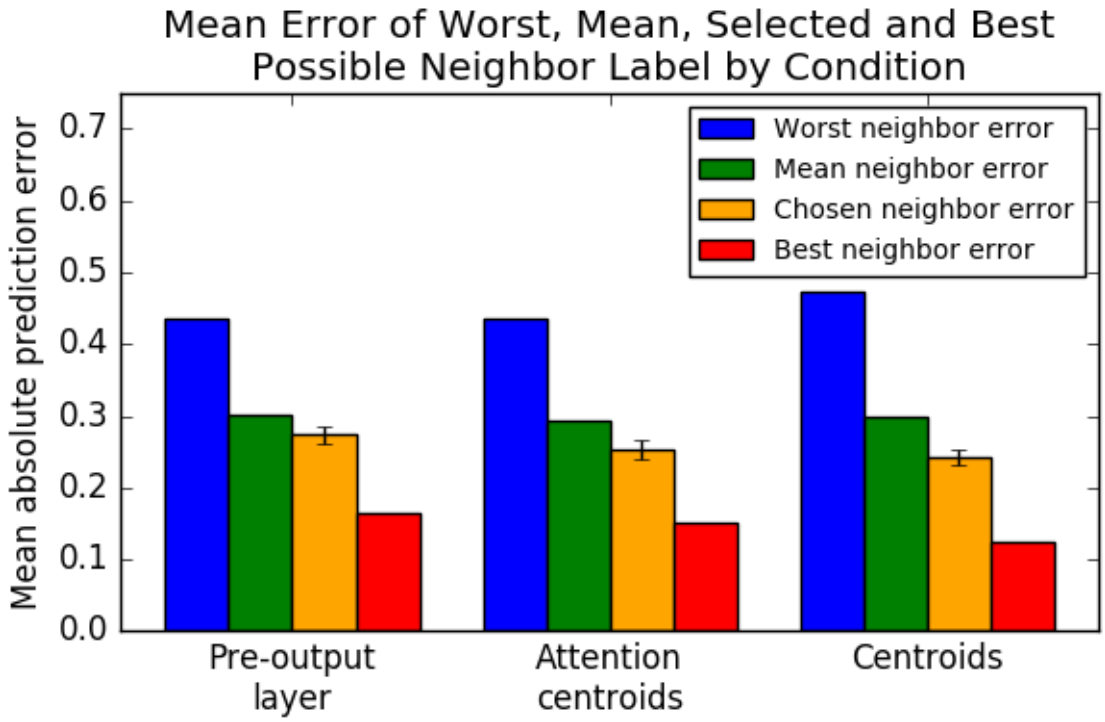

Figure 5.7: Error of the true toxicity of the chosen neighbor in the prediction task, compared to the best, worst and mean error.

toxicity error than their own baseline human accuracy.

5.4 Discussion

This chapter describes a user study which tested the relative merits of three explanatory algorithms (pre-output layer, attention centroids and centroids) in terms of several human outcomes. It found that human subjects, when asked to select neighbors to serve as evidence in making predictions about the items of interest, preferred the attention centroids algorithm by a large margin.

However, none of the three algorithms were able to actually improve human performance on this prediction task, a goal which still remains elusive in the literature on interpretable machine learning. As Table 5.2 shows, the best neighbor from each

	Neighbor error			
	Worst	Mean	Selected	Best
Pre-output layer	0.436	0.301	0.274	0.166
Attention centroids	0.436	0.293	0.254	0.15
Centroids	0.473	0.299	0.243	0.125

Table 5.2: Error of chosen neighbor across retrieval algorithms.

algorithm was generally closer to the true label than the baseline human prediction, but subjects were not able to select this optimal neighbor consistently enough for either their mean neighbor selection nor their mean final error to be any lower than their baseline accuracy.

The slight edge in performance enjoyed by the centroids method in terms of selected neighbor accuracy and human prediction error change implies that neighbor diversity may be an important design factor in this type of explanation. Showing neighbors that represent a wider range of possible outcomes appears to make it easier to select a neighbor that provides real insight into the decision at hand. The idea of diversity has a long history in the information retrieval literature, and it is possible that concepts like maximum marginal relevance (*Carbonell and Goldstein*, 1998) could be fruitfully applied to this task.

Another concern in the task of retrieving explanatory examples is that of true label variance. Comments in the *Wulczyn et al.* (2017) dataset were given a binary label by roughly 10 crowd workers each, and we use the mean of these binary responses as the true labels in the training and explanation of our model. What that means is that a comment like figure 4.6.1, for which the true label was 0.5, could easily have had a different label depending on which workers annotated it. We attempt to reduce this variance in the user study by collecting 50 binary responses for each item-of-interest, but the true labels of the neighbors we present still have this property.

Even less subjective tasks will often have this property of a nondeterministic

relationship between input and true label. For example, in the fake review task studied by *Lai and Tan* (2019), two reviews could have identical texts but opposite labels. The truthfulness or fakeness of a review is a property external to the text itself which is associated with certain text qualities but not defined by them. This kind of task will lead to the same sort of true label variance as in the *Wulczyn et al.* (2017) dataset.

Because of this variance, the premise of displaying a small number of carefully-chosen evidence comments for a user to apply analogical reasoning to may be the wrong approach entirely. It is possible that what is really needed is a an interactive interface like that proposed in *Ming et al.* (2018), using a dimension reduction technique such as T-SNE (*Maaten and Hinton*, 2008) to display a large number of nearest neighbors in a way that conveys the level of support for a given true label amongst the neighbors on display. Such a system would be more robust to true label variance, as it would display aggregate labels across clusters of similar neighbors.

5.4.1 Hybrid robustness

The study described in this chapter represents a second attempt to realize the core goal of this dissertation and, I argue, the core goal of the contemporary interpretability literature: that of using explanations to improve hybrid human-machine performance on a decision task.

We do not succeed in this primary objective. We find that subjects were only somewhat able to select apt neighbors to identify as evidence, and having done so their performance actually dropped on average. So the theory engendered by the result of the Chapter III study, that examples might succeed where attribution failed, was not supported by the results of this study.

However, given how large the design space is for example-based explanations and how little attention has been given to them (especially their evaluation), and how

difficult the research community has found it so far to demonstrate a positive effect from explanations of any kind, it would have been surprising if we had achieved a significant positive result in this study. As it is, by overcoming the basic barrier of producing text examples which are simple enough to read and comprehend, we were able to expose specific design issues that could inform future work on a similar topic, such as the importance of neighbor diversity.

CHAPTER VI

Conclusion

This dissertation proposes that the primary goal of interpretable machine learning is to improve the quality of decisions made by humans in the presence of AI models. I focus on the local robustness use case, in which human analysts attempt to make decisions in the presence of model advice and must decide from case to case whether to accept or override that advice.

To that end, I introduce two algorithms for local explanations of neural net text classifiers: adversarial attention for feature attribution, and attention-conscious retrieval of explanatory examples. Each algorithm is evaluated by both a functional empirical evaluation and a rigorous application-based user study. Both types of work presented in this document, methods and evaluation, represent contributions to the overall literature on interpretability and suggest future work in the field.

6.1 Methods

The two algorithms introduced in this work seek to generate explanations which leverage respectively the two dimensions of the data matrix that any machine learning algorithm operates on: features (columns) and examples (rows).

The adversarial attention method described in Chapter II uses an adversarial training regime to combine the idea of model attention with the idea of counterfactual

reasoning used by many posthoc methods (e.g *Mothilal et al.* (2019); *Li et al.* (2016); *Wachter et al.* (2017b)). It shifts the counterfactual logic of evaluating "how would the model output change if this feature changed" from posthoc analysis to training, producing a model whose attention mask consists of all tokens that are liable to provide predictive signal about the outcome.

This method comes at a time when there is controversy in the interpretability literature about whether model attention can be treated as explanations for model behavior (*Jain and Wallace*, 2019; *Serrano and Smith*, 2019; *Vashishth et al.*, 2019; *Wiegreffe and Pinter*, 2019). *Jain and Wallace* (2019), for example, find that the attribution masks produced by a standard attention mechanism show poor agreement with other attribution methods. While the empirical evaluation presented in Chapter II does not include this exact analysis, the fact that our method competes favorably with methods like LIME (*Ribeiro et al.*, 2016) is an indication that it might represent a solution to some of the problems that have been discussed in this series of papers.

The explanatory example method described in Chapter IV uses a variant of the Chapter II attention method to simplify the representations of input texts in order to then retrieve and display explanatory examples from the training data, for the purpose of helpfully contextualizing model predictions. The contribution of this algorithm is to articulate a minimal set of criteria (relevance and fidelity) for generating such explanatory examples, and proposing a method designed to optimize a balance of these two qualities.

This contribution is made particularly strong because of the relative paucity of literature pertaining to this type of explanation. Existing work has tended to be focused on image rather than text data and on the utility of explanatory examples for non-human-oriented goals, and accordingly has not tended to include user studies measuring behavioral outcomes (e.g. *Papernot and McDaniel* (2018); *Wallace et al.* (2018); *Kim et al.* (2016); *Cai et al.* (2019a)). Our algorithm contributes to this

literature by producing examples which are actually intended for human consumption in addition to holding other types of utility.

6.1.1 Intrinsic versus posthoc interpretability

A dichotomy in the current interpretability literature that I discuss briefly in Chapter I is the tension between intrinsic interpretability methods like neural attention and posthoc methods like LIME (*Ribeiro et al.*, 2016). The methods proposed in this dissertation are a reflection of this tension. The Chapter II attention mechanism is an intrinsic interpretability method, while the Chapter IV example method is a posthoc method.

By forcing complex models to be interpretable, we not only end up with explanations we can use to reason about the behavior of those models, but with models that reason in a more interpretable way. The adversarial attention mechanism described in Chapter II teaches the classifier to generate attention masks by reasoning "this is the minimum set of words which, if removed from this text, would render it nontoxic". This training regime produces attention masks with a more semantically precise definition than the more conventional attention objective of "the minimum set of words needed to make an accurate prediction", which has been noted to produce masks that accord poorly with human judgement (*Feng et al.*, 2018; *Jain and Wallace*, 2019).

By contrast, the explanatory example retrieval algorithm described in Chapter IV and then evaluated in Chapter V is a posthoc explanation mechanism because it explains the behavior of an existing model using a retroactive process (retrieving and presenting explanatory examples). However, the only way that example-based explanations could be generated in a way *intrinsic* to the functionality of a classifier is if the classifier itself were example-based, such as the popular k-nearest neighbors algorithm. This type of model, while popular in application, tends to suffer from generalization problems (*Hastie et al.*, 2001b) and is not regarded as an especially

active area of machine learning research. For example, the proceedings of the 2019 International Conference in Machine Learning (ICML) contain 74 instances of the word "neural" and only 4 of the word "neighbor"[1].

So there is a conflict between a desire for models with intrinsic interpretability and a desire for more work on example-based explanations. Generalization-based (i.e. model-based) machine learning is preferred to example-based machine learning, but it is impossible to generate example-based explanations which are perfectly coupled to generalization-based predictions in the way that is called for by the idea of intrinsic interpretability.

A partial solution to this paradox exists in ideas like conformal prediction, in which a generalization-based model still makes predictions, but these predictions are either nudged through optimization (e.g. *Chen et al.* (2018)), or adjusted in a posthoc manner (e.g. *Papernot and McDaniel* (2018)) to conform to their neighbors within the representation space learned by the model. I believe that the former approach represents a rich avenue for further research, with the potential to generate models that combine the power of generalization with the interpretability of example-based methods.

One way to perform this nudging could be the inclusion of a generative objective in the training process for the classifier. We can place a generative objective on the output layer of a classification model such as an attention LSTM which encourages it to not only make accurate predictions but also be able to regenerate its (attention-masked) input. If done properly, I hypothesize that this would produce a model whose pre-output layer representation is useful for both prediction and retrieval of semantically similar items (as opposed to the structural similarity we noted in section 4.3).

The idea of combining generative and discriminative objectives accords with recent

[1] https://icml.cc/Conferences/2019/Schedule?type=Poster

developments in NLP that have demonstrated the utility of pretraining a generative language model on unlabeled data and then fine-tuning such a model for particular tasks (*Peters et al.*, 2018; *Devlin et al.*, 2018). Starting with one of these models would be a practical way of approaching this joint optimization idea.

The Chapter II adversarial attention algorithm could also potentially benefit from the inclusion of generative modeling. This algorithm requires a special training paradigm to prevent the adversarial learner from learning the relationship between the presence of masking and the true toxicity label of the comment. This could be avoided if, rather than strictly blanking out tokens with low attention weights, the model were to instead replace them with manufactured tokens chosen specifically to fool the adversarial predictor. This would obviate the need for the confusion step of model training, and would additionally mitigate any incidental data leakage that might be slipping through that step.

6.1.2 Interpretability and active learning

One advantage of model attention over other feature attribution techniques is that it produces an attention mask as an additional output that can be manipulated via its own terms in the model objective function. In the adversarial attention model described in Chapter II, we optimize this attention mask to be sparse but comprehensive, as well as cohesive.

This property could be used to close the loop on interpretability and active learning. If known model errors are shown to humans along with their corresponding attention masks, those humans can potentially diagnose those errors at the attention level by identifying tokens the model should or should not have been attending to. These attention-level corrections can then be fed back into the model training as a target value on the attention output for those tokens.

Combining model attention and active learning in this way could represent a

breakthrough in model debugging, as it would allow precisely targeted human feedback into a buggy model without being dependent on the difficult task of selecting additional data to label, as in traditional active learning approaches (*Settles*, 2010).

6.2 Evaluation

In Chapter I, I argue that the goal of improving the predictive performance of human decision-makers in conjunction with AI models, what can be described as a hybrid system drawing on both human and model insight, is the "holy grail" of interpretability work that no study has been able to conclusively claim so far.

Every evaluation performed in this dissertation is a reflection of this basic goal in some way. The two empirical evaluations represent operationalizations of qualities we theorize to be important in utilizing the corresponding methods for effective decision-making (completeness and analogical validity, respectively), while the two user studies are direct implementations of this goal of increasing robustness.

While both user studies have primarily negative results, I argue that this result does not invalidate this primary claim. In combination with other similar recent works, it suggests that the task of improving hybrid robustness is extremely challenging from both a machine learning and a human-computer interaction perspective. Furthermore, because the bulk of the work on this problem has come from the machine learning community, there is a large expectation gap in the field, in which the topic has attracted a great deal of attention in the methods literature but has proven to be consistently unhelpful in human performance outcomes in the limited human experimentation literature that has been published (e.g. *Lage et al.* (2018); *Lai and Tan* (2019); *Poursabzi-Sangdeh et al.* (2018)).

This dissertation works to characterize if not close this gap by including two relatively large-scale user studies assessing the impact of attribution-based and example-based explanations respectively on human performance on a reasonably realistic pre-

diction task. While we are as unable as the works cited above to achieve a robustness improvement, these two studies help push out our understanding of the design space of applied interpretability.

From the feature-based study described in Chapter III, for example, we learn that while feature highlighting does not improve human accuracy, it does change the distribution of human error, causing subjects to make more false negatives but fewer false positives. Noting this, I suggest that future work in applied interpretability should be conscious of what type of error it seeks to reduce in its subjects. For example, a model intended to be used in an applied interpretability application could actually be tuned to produce more false positives than false negatives on the basis that this type of error seems easier to overturn with the benefit of explanations. At the very least, designers should carefully consider the prevalence and severity of different types of errors in choosing whether to incorporate explanations into their applications.

More generally, these ambiguous results begin to suggest that the relationship between the baseline distributions of human and model error is a huge factor in determining whether interpretability can aid human performance on a given decision task. The less that either type of agent can uniquely contribute to the joint system, the less potential there is for improvement, and the less likely it is that the combined human-model decision system can outperform humans or models alone.

We address this issue in both user studies. In the Chapter III study, we oversample model errors from across the full range of true toxicity scores in the dataset, resulting in a comment sample for which the model and baseline human performance is similar (accuracy 0.4 versus 0.5). In the Chapter V study we sample comments from those with borderline predicted toxicity, on which the model is naturally somewhat unreliable. This again results in a comparable model versus human baseline performance (mean absolute error 0.24 versus 0.21).

While these are both reasonable solutions to the problem of unbalanced model and human error distributions, future evaluation studies may need to include some sort of "potential improvement assessment" pilot in which baseline human performance is measured and the complementarity of human and model error on that particular decision task is assessed to see what improvement could come from better human understanding of model behavior. As the literature matures, this assessment step could grow to accommodate a more nuanced understanding of the types of error interpretability is liable to mitigate (like the type I/type II distinction we noticed in our first user study).

It is possible that beyond understanding the *distribution* of human error, it may be necessary to understand the *reasons* for human error on these decision tasks before we can design interpretability methods to mitigate those errors. One way to collect this data in future evaluation studies would be to ask human subjects to explain their decisions by selecting areas of text (or images, or tabular data, as appropriate), and then diagnosing human errors at the feature level by comparing the annotations of accurate subjects with inaccurate ones. Gaze studies might be even more fruitful for capturing the true reasons underlying human decisions (and thus human errors).

We find in the Chapter V study that subjects struggle to choose "good" neighbors and to use these neighbors to make better assessments about toxicity. There are a few possible reasons for this failure, including:

- Diversity of presented neighbors (hinted at by the slightly better performance of the centroids method in this regard)
- Unreliability of ground truth labels for those neighbors (because of label noise in the *Wulczyn et al.* (2017) dataset)
- Risk of presenting neighbors that are actually outliers (exacerbated in our study by choosing the closest neighbor of each possible class)

These are all issues that future studies on this topic need to explore, but it was

only by testing an algorithm which fulfilled the basic criteria of producing neighbors that were both visually recognizably relevant and pertinent to the model's prediction on each item-of-interest that these deeper design issues could have been revealed.

Beyond the specific issues that we identified as salient, example-based explanations represent a vast, untapped design space. Some of the design decisions made in the Chapter V study include: 1) how to select examples; 2) which examples and how many examples to display to users; 3) what information to display about each example; 4) when (i.e. what order) to display each type of information; 5) how to account for label noise and/or labeler disagreement in the presence of training examples; 6) whether to display synthetic examples generated by the model or true examples drawn from the dataset; 7) whether and how to integrate example-based and feature-based explanations. While this list is by no means exhaustive, it helps illustrate the combinatorial explosion of options for how to combine designs elements in this type of explanatory system.

Further experimentation may be able to arrive at an optimal combination of predictive model, neighbor retrieval algorithm, and visual presentation that gives human users ways to reason about the reliability of model predictions in a way that actually improves performance above both baseline human and model performance. The Chapter V user study represents a probe into this space which resolves one basic unaddressed problem in the literature (that of retrieving comprehensible neighbors with similar model behavior), while revealing a number of secondary design issues for this type of explanation method (choice diversity, label noise, neighbor representativeness).

Therefore, I argue that the two user studies presented in this dissertation are valuable contributions to the interpretability literature as a whole because they help highlight this expectation gap between method and application, and because they suggest methodological research directions that would not have been evident without

their conclusions.

6.2.1 Training effects

Experiments in applied interpretability have tended to be one-shot prediction tasks that implicitly assume that the human and model agents in question are static entities with set predefined prediction and error behavior. While a model can always be retrained with different hyperparameters (e.g. a higher penalty on false negatives), human agents are also dynamic entities capable of learning and adapting to the capabilities of a system.

That is to say, it is possible that the desired performance improvement might emerge in longer-term training of human subjects over the simple one-shot experiments that have been performed so far. With real-time feedback about the correctness of their predictions, users might get better and better at the given task (toxicity prediction in our case). It is possible that explanations would have an impact on this training process, either on the time required to achieve a certain level of improvement or on the level of improvement attainable.

6.3 Conclusion

In this work I present two novel interpretability algorithms which fill significant gaps in the interpretability methods literature and two rigorous user studies which address calls for more rigorous user testing of interpretability concepts while revealing insights about the human factors involved in machine learning interpretability. These contributions reveal a number of potential avenues for future work on both methods and human experimentation.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abbott, R., B. Ecker, P. Anand, and M. Walker (2016), Internet Argument Corpus 2.0: An SQL Schema for Dialogic Social Media and the Corpora to Go With It, in *Language Resources and Evaluation Conference*.

Abdul, A., J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli (2018), Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–18, ACM Press, Montreal QC, Canada, doi: 10.1145/3173574.3174156.

Ancona, M., E. Ceolini, C. ztireli, and M. Gross (2018), Towards better understanding of gradient-based attribution methods for Deep Neural Networks, in *arXiv:1711.06104 [cs, stat]*, arXiv: 1711.06104.

Anderson, A. A., D. Brossard, D. A. Scheufele, M. A. Xenos, and P. Ladwig (2014), The Nasty Effect: Online Incivility and Risk Perceptions of Emerging Technologies: Crude comments and concern, *Journal of Computer-Mediated Communication*, *19*(3), 373–387, doi: 10.1111/jcc4.12009.

Anderson, A. A., S. K. Yeo, D. Brossard, D. A. Scheufele, and M. A. Xenos (2016), Toxic Talk: How Online Incivility Can Undermine Perceptions of Media, *International Journal of Public Opinion Research*, doi: 10.1093/ijpor/edw022.

Ardila, D., et al. (2019), End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography, *Nature Medicine*, *25*(6), 954–961, doi: 10.1038/s41591-019-0447-x.

Arora, S., Y. Liang, and T. Ma (2017), A Simple but Tough-to-Beat Baseline for Sentence Embeddings, in *Proceedings of the International Conference on Learning Representations*.

Arras, L., F. Horn, G. Montavon, K.-R. Mller, and W. Samek (2017), "What is relevant in a text document?": An interpretable machine learning approach, *PLOS ONE*, *12*(8), e0181,142, doi: 10.1371/journal.pone.0181142.

Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Mller, and W. Samek (2015), On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation, *PLOS ONE*, *10*(7), e0130,140, doi: 10.1371/journal.pone.0130140, 00034.

Bansal, G., B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz (2019), Updates in Human-AI Teams: Understanding and Addressing the Performance/Compatibility Tradeoff, *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 2429–2437, doi: 10.1609/aaai.v33i01.33012429.

Binns, R., M. Veale, M. Van Kleek, N. Shadbolt, M. Veale, M. Van Kleek, and N. Shadbolt (2017), Like Trainer, Like Bot? Inheritance of Bias in Algorithmic Content Moderation, in *Social Informatics*, vol. 10540, pp. 405–415, Springer, Cham, doi: 10.1007/978-3-319-67256-4_32.

Blackwell, L., T. Chen, S. Schoenebeck, and C. Lampe (2018a), When Online Harassment is Perceived as Justified, in *Proceedings of the Twelfth International AAAI Conference on Web and Social Media*.

Blackwell, L., J. Dimond, S. Schoenebeck, and C. Lampe (2018b), Classification and Its Consequences for Online Harassment: Design Insights from HeartMob, in *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2018)*, pp. 1–19.

Bussone, A., S. Stumpf, and D. O'Sullivan (2015), The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems, in *2015 International Conference on Healthcare Informatics*, pp. 160–169, doi: 10.1109/ICHI.2015.26.

Cai, C. J., J. Jongejan, and J. Holbrook (2019a), The effects of example-based explanations in a machine learning interface, in *Proceedings of the 24th International Conference on Intelligent User Interfaces - IUI '19*, pp. 258–262, ACM Press, Marina del Ray, California, doi: 10.1145/3301275.3302289.

Cai, C. J., et al. (2019b), Human-Centered Tools for Coping with Imperfect Algorithms During Medical Decision-Making, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, pp. 1–14, ACM Press, Glasgow, Scotland Uk, doi: 10.1145/3290605.3300234.

Carbonell, J., and J. Goldstein (1998), The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries, in *Proceedings of the international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 335–336.

Carton, S., Q. Mei, and P. Resnick (2018), Extractive Adversarial Networks: High-Recall Explanations for Identifying Personal Attacks in Social Media Posts, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, arXiv: 1809.01499.

Carton, S., et al. (2016), Identifying Police Officers at Risk of Adverse Events, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp. 67–76, ACM Press, San Francisco, California, USA, doi: 10.1145/2939672.2939698.

Caruana, R., H. Kangarloo, J. D. Dionisio, U. Sinha, and D. Johnson (1999), Case-based explanation of non-case-based learning methods., *Proceedings of the AMIA Symposium*, pp. 212–215, 00007.

Chancellor, S., Y. Kalantidis, J. A. Pater, M. De Choudhury, and D. A. Shamma (2017), Multimodal Classification of Moderated Online Pro-Eating Disorder Content, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 3213–3226, ACM, New York, NY, USA, doi: 10.1145/3025453.3025985.

Chandrasekharan, E., M. Samory, S. Jhaver, H. Charvat, A. Bruckman, C. Lampe, J. Eisenstein, and E. Gilbert (2018), The Internet's Hidden Rules: An Empirical Study of Reddit Norm Violations at Micro, Meso, and Macro Scales, *Proceedings of the ACM on Human-Computer Interaction - CSCW*, *2*(CSCW), 1–25, doi: 10.1145/3274301.

Chen, C., O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin (2018), This Looks Like That: Deep Learning for Interpretable Image Recognition, *arXiv:1806.10574 [cs, stat]*, arXiv: 1806.10574.

Cheng, J., C. Danescu-Niculescu-Mizil, and J. Leskovec (2015), Antisocial Behavior in Online Discussion Communities, in *Proceedings of the International Conference on Web and Social Media*, p. 10.

Dastin, J. (2018), Amazon scraps secret AI recruiting tool that showed bias against women, *Reuters*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv:1810.04805 [cs]*, arXiv: 1810.04805.

DiFranzo, D., S. H. Taylor, F. Kazerooni, O. D. Wherry, and N. N. Bazarova (2018), Upstanding by Design: Bystander Intervention in Cyberbullying, in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pp. 1–12, ACM Press, Montreal QC, Canada, doi: 10.1145/3173574.3173785.

Doshi-Velez, F., and B. Kim (2017), Towards A Rigorous Science of Interpretable Machine Learning, *arXiv:1702.08608 [cs, stat]*, arXiv: 1702.08608.

Ehsan, U., B. Harrison, L. Chan, and M. O. Riedl (2018), Rationalization: A Neural Machine Translation Approach to Generating Natural Language Explanations, in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, arXiv: 1702.07826.

Eslami, M., S. R. Krishna Kumaran, C. Sandvig, and K. Karahalios (2018), Communicating Algorithmic Process in Online Behavioral Advertising, in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 432:1–432:13, ACM, New York, NY, USA, doi: 10.1145/3173574.3174006, event-place: Montreal QC, Canada.

Feng, S., E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber (2018), Pathologies of Neural Models Make Interpretations Difficult, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, arXiv: 1804.07781.

Fiesler, C., J. A. Jiang, J. McCann, K. Frye, and J. R. Brubaker (2018), Reddit Rules! Characterizing an Ecosystem of Governance, in *Twelfth International AAAI Conference on Web and Social Media*.

Finkel, N. J. (1989), The Insanity Defense Reform Act of 1984: Much ado about nothing, *Behavioral Sciences & the Law*, *7*(3), 403–419, doi: 10.1002/bsl.2370070309.

for Computing Machinery (ACM), A. (2016), *CyberSafety'16: Proceedings of the First International Workshop on Computational Methods for CyberSafety*, ACM, New York, NY, USA.

Fortuna, P., and S. Nunes (2018), A Survey on Automatic Detection of Hate Speech in Text, *ACM Computing Surveys*, *51*(4), 1–30, doi: 10.1145/3232676.

Friedler, S. A., C. D. Roy, C. Scheidegger, and D. Slack (2019), Assessing the Local Interpretability of Machine Learning Models, *arXiv:1902.03501 [cs, stat]*, arXiv: 1902.03501.

Galassi, A., M. Lippi, and P. Torroni (2019), Attention, please! A Critical Review of Neural Attention Models in Natural Language Processing, *arXiv:1902.02181 [cs, stat]*, arXiv: 1902.02181.

Ghorbani, A., J. Wexler, J. Zou, and B. Kim (2019), Towards Automatic Concept-based Explanations, *arXiv:1902.03129 [cs, stat]*, arXiv: 1902.03129.

Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal (2018), Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning, *arXiv:1806.00069 [cs, stat]*, arXiv: 1806.00069.

Goel, A., and B. Diaz-Agudo (2017), What's Hot in Case-Based Reasoning, in *Thirty-First AAAI Conference on Artificial Intelligence*.

Golbeck, J., et al. (2017), A Large Labeled Corpus for Online Harassment Research, in *Proceedings of the 2017 ACM on Web Science Conference*, pp. 229–233, ACM Press, doi: 10.1145/3091478.3091509.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014), Generative Adversarial Nets, in *Advances in Neural Information Processing Systems*, pp. 2672–2680.

Guidotti, R., A. Monreale, F. Turini, and D. Pedreschi (2018), A Survey Of Methods For Explaining Black Box Models, *arXiv preprint arXiv:1802.01933*.

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger (2017), On Calibration of Modern Neural Networks, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330, JMLR.org, event-place: Sydney, NSW, Australia.

Hastie, T., J. Friedman, and R. Tibshirani (2001a), Introduction, in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, edited by T. Hastie, J. Friedman, and R. Tibshirani, Springer Series in Statistics, pp. 1–8, Springer New York, New York, NY, doi: 10.1007/978-0-387-21606-5_1.

Hastie, T., R. Tibshirani, and J. Friedman (2001b), *The Elements of Statistical Learning*, Springer Series in Statistics, second ed., Springer, New York.

Hosseini, H., S. Kannan, B. Zhang, and R. Poovendran (2017), Deceiving Google's Perspective API Built for Detecting Toxic Comments, *arXiv:1702.08138 [cs]*, arXiv: 1702.08138.

Hosseinmardi, H., S. A. Mattson, R. Ibn Rafiq, R. Han, Q. Lv, and S. Mishra (2015), Analyzing Labeled Cyberbullying Incidents on the Instagram Social Network, in *Social Informatics*, edited by T.-Y. Liu, C. N. Scollon, and W. Zhu, Lecture Notes in Computer Science, pp. 49–66, Springer International Publishing.

Jain, S., and B. C. Wallace (2019), Attention is not Explanation, *arXiv:1902.10186 [cs]*, arXiv: 1902.10186.

Kay, M., S. N. Patel, and J. A. Kientz (2015), How Good is 85%?: A Survey Tool to Connect Classifier Evaluation to Acceptability of Accuracy, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pp. 347–356, ACM Press, Seoul, Republic of Korea, doi: 10.1145/2702123.2702603.

Keane, M. T., and E. M. Kenny (2019), How Case Based Reasoning Explained Neural Networks: An XAI Survey of Post-Hoc Explanation-by-Example in ANN-CBR Twins, *arXiv:1905.07186 [cs]*, arXiv: 1905.07186.

Kennedy, G., A. McCollough, E. Dixon, A. Bastidas, J. Ryan, C. Loo, and S. Sahay (2017), Technology Solutions to Combat Online Harassment, in *Proceedings of the First Workshop on Abusive Language Online*, pp. 73–77, Association for Computational Linguistics, doi: 10.18653/v1/W17-3011.

Kim, B., R. Khanna, and O. O. Koyejo (2016), Examples are not enough, learn to criticize! Criticism for Interpretability, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, pp. 2280–2288, Curran Associates, Inc.

Kim, B., M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres (2017), Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV), *arXiv:1711.11279 [stat]*, arXiv: 1711.11279.

Kingma, D. P., and J. Ba (2014), Adam: A Method for Stochastic Optimization, *Proceedings of the 3rd International Conference on Learning Representations*.

Kleinberg, J., H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan (2017), Human Decisions and Machine Predictions, *The Quarterly Journal of Economics*, doi: 10.1093/qje/qjx032.

Koh, P. W., and P. Liang (2017), Understanding Black-box Predictions via Influence Functions, *arXiv:1703.04730 [cs, stat]*, arXiv: 1703.04730.

Kumar, R., A. K. Ojha, M. Zampieri, and S. Malmasi (2018), Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Lage, I., E. Chen, J. He, M. Narayanan, B. Kim, S. Gershman, and F. Doshi-Velez (2018), An Evaluation of the Human-Interpretability of Explanation, in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*.

Lai, V., and C. Tan (2019), On Human Predictions with Explanations and Predictions of Machine Learning Models: A Case Study on Deception Detection, in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, p. 17.

Lakkaraju, H., S. H. Bach, and J. Leskovec (2016), Interpretable Decision Sets: A Joint Framework for Description and Prediction, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp. 1675–1684, ACM Press, San Francisco, California, USA, doi: 10.1145/2939672.2939874.

Lakkaraju, H., E. Kamar, R. Caruana, and J. Leskovec (2017), Interpretable & Explorable Approximations of Black Box Models, *arXiv:1707.01154 [cs]*, arXiv: 1707.01154.

Lei, T., R. Barzilay, and T. Jaakkola (2016), Rationalizing Neural Predictions, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117.

Lenhart, A., M. Ybarra, K. Zickuhr, and M. Price-Feeney (2016), Online Harassment, Digital Abuse, and Cyberstalking in America, *Tech. rep.*, Data & Society Research Institute.

Li, J., W. Monroe, and D. Jurafsky (2016), Understanding Neural Networks through Representation Erasure, *arXiv preprint arXiv:1612.08220*.

Li, O., H. Liu, C. Chen, and C. Rudin (2017), Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions, *arXiv:1710.04806 [cs, stat]*, arXiv: 1710.04806.

Link, D., and B. Hellingrath (2016), A Human-is-the-Loop Approach for Semi-Automated Content Moderation, in *Proceedings of ISCRAM*, p. 13.

Lipton, Z. C., et al. (2016), The Mythos of Model Interpretability, *IEEE Spectrum*.

Lundberg, S. M., and S.-I. Lee (2017), A Unified Approach to Interpreting Model Predictions, *Advances in Neural Information Processing Systems*, p. 10.

Luong, T., H. Pham, and C. D. Manning (2015), Effective Approaches to Attention-based Neural Machine Translation, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Association for Computational Linguistics, Lisbon, Portugal, doi: 10.18653/v1/D15-1166.

Maaten, L. v. d., and G. Hinton (2008), Visualizing data using t-SNE, *Journal of Machine Learning Research*, *9*(Nov), 2579–2605, 01822.

Malmasi, S., and M. Zampieri (2018), Challenges in discriminating profanity from hate speech, *Journal of Experimental & Theoretical Artificial Intelligence*, *30*(2), 187–202, doi: 10.1080/0952813X.2017.1409284.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013), Distributed representations of words and phrases and their compositionality, in *Advances in neural information processing systems*, pp. 3111–3119.

Ming, Y., H. Qu, and E. Bertini (2018), RuleMatrix: Visualizing and Understanding Classifiers with Rules, *arXiv:1807.06228 [cs, stat]*, arXiv: 1807.06228.

Mohseni, S., and E. D. Ragan (2018), A Human-Grounded Evaluation Benchmark for Local Explanations of Machine Learning, *arXiv:1801.05075 [cs]*, arXiv: 1801.05075.

Mothilal, R. K., A. Sharma, and C. Tan (2019), Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations, *arXiv:1905.07697 [cs, stat]*, arXiv: 1905.07697.

Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu (2019), Interpretable machine learning: definitions, methods, and applications, *arXiv preprint arXiv:1901.04592*.

Napoles, C., J. Tetreault, A. Pappu, E. Rosato, and B. Provenzale (2017), Finding Good Conversations Online: The Yahoo News Annotated Comments Corpus, in *Proceedings of the 11th Linguistic Annotation Workshop*, doi: 10.18653/v1/W17-0802.

Narayanan, M., E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez (2018), How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation, *arXiv preprint arXiv:1802.00682*.

Nguyen, D. (2018), Comparing Automatic and Human Evaluation of Local Explanations for Text Classification, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1069–1078.

Nobata, C., J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang (2016), Abusive Language Detection in Online User Content, in *Proceedings of the 25th International Conference on World Wide Web*, pp. 145–153.

Olteanu, A., K. Talamadupula, and K. R. Varshney (2017), The Limits of Abstract Evaluation Metrics: The Case of Hate Speech Detection, in *Proceedings of the 2017 ACM on Web Science Conference - WebSci '17*, pp. 405–406, ACM Press, Troy, New York, USA, doi: 10.1145/3091478.3098871.

Papadopoulos, H. (2008), Inductive Conformal Prediction: Theory and Application to Neural Networks, in *Tools in Artificial Intelligence*, edited by P. Fritzsche, InTech, doi: 10.5772/6078.

Papernot, N., and P. McDaniel (2018), Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning, *arXiv:1803.04765 [cs, stat]*, arXiv: 1803.04765.

Pavlopoulos, J., P. Malakasiotis, and I. Androutsopoulos (2017), Deep Learning for User Comment Moderation, in *Proceedings of the First Workshop on Abusive Language Online*, pp. 25–35, doi: 10.18653/v1/W17-3004.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018), Deep contextualized word representations, *arXiv:1802.05365 [cs]*, arXiv: 1802.05365.

Pew (2016), The Political Environment on Social Media, *Tech. rep.*, Pew Research Center.

Poursabzi-Sangdeh, F., D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach (2018), Manipulating and Measuring Model Interpretability, *arXiv:1802.07810 [cs]*, arXiv: 1802.07810.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016), "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*, pp. 1135–1144, ACM, doi: 10.1145/2939672.2939778.

Ribeiro, M. T., S. Singh, and C. Guestrin (2018), Anchors: High Precision Model-Agnostic Explanations, in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, p. 9.

Ross, A. S., M. C. Hughes, and F. Doshi-Velez (2017), Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations, *arXiv preprint arXiv:1703.03717*, 00000.

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

Salmon, M. H. (2012), *Introduction to Logic and Critical Thinking*, Cengage Learning.

Samek, W., T. Wiegand, and K.-R. Mller (2017), Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models, *arXiv:1708.08296 [cs, stat]*, arXiv: 1708.08296.

Serrano, S., and N. A. Smith (2019), Is Attention Interpretable?, *arXiv:1906.03731 [cs]*, arXiv: 1906.03731.

Settles, B. (2010), Active Learning Literature Survey, *Tech. rep.*, University of Wisconsin-Madison.

Shafer, G., and V. Vovk (2007), A tutorial on conformal prediction, *arXiv:0706.3188 [cs, stat]*, arXiv: 0706.3188.

Shrikumar, A., P. Greenside, A. Shcherbina, and A. Kundaje (2016), Not Just a Black Box: Learning Important Features Through Propagating Activation Differences, *arXiv:1605.01713 [cs]*, arXiv: 1605.01713.

Shrikumar, A., P. Greenside, and A. Kundaje (2017), Learning Important Features Through Propagating Activation Differences, *arXiv:1704.02685 [cs]*, arXiv: 1704.02685.

Simonyan, K., A. Vedaldi, and A. Zisserman (2013), Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, *arXiv:1312.6034 [cs]*, arXiv: 1312.6034.

Springer, A., V. Hollis, and S. Whittaker (2017), Dice in the Black Box: User Experiences with an Inscrutable Algorithm, in *2017 AAAI Spring Symposium Series*.

Stenetorp, P., S. Pyysalo, G. Topi, T. Ohta, S. Ananiadou, and J. Tsujii (2012), BRAT: a web-based tool for NLP-assisted text annotation, in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107.

Sutskever, I., O. Vinyals, and Q. V. Le (2014), Sequence to sequence learning with neural networks, in *Advances in neural information processing systems*, pp. 3104–3112, 00938.

Svec, A., M. Pikuliak, M. Simko, and M. Bielikova (2018), Improving Moderation of Online Discussions via Interpretable Neural Models, in *Proceedings of the 2nd Workshop on Abusive Language Online*.

Vashishth, S., S. Upadhyay, G. S. Tomar, and M. Faruqui (2019), Attention Interpretability Across NLP Tasks, *arXiv:1909.11218 [cs]*, arXiv: 1909.11218.

Wachter, S., B. Mittelstadt, and L. Floridi (2017a), Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation, *International Data Privacy Law*, *7*(2), 76–99, doi: 10.1093/idpl/ipx005.

Wachter, S., B. Mittelstadt, and C. Russell (2017b), Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR, *SSRN Electronic Journal*, doi: 10.2139/ssrn.3063289.

Wallace, E., S. Feng, and J. Boyd-Graber (2018), Interpreting Neural Networks With Nearest Neighbors, *arXiv:1809.02847 [cs]*, arXiv: 1809.02847.

Wang, C. (2018), Interpreting Neural Network Hate Speech Classiers, in *Proceedings of the 2nd Workshop on Abusive Language Online*, p. 7.

Warshaw, J., T. Matthews, S. Whittaker, C. Kau, M. Bengualid, and B. A. Smith (2015), Can an Algorithm Know the "Real You"?: Understanding People's Reactions to Hyper-personal Analytics Systems, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 797–806, ACM, New York, NY, USA, doi: 10.1145/2702123.2702274, event-place: Seoul, Republic of Korea.

Waseem, Z., W. H. K. Chung, D. Hovy, and J. Tetreault (Eds.) (2017a), *Proceedings of the First Workshop on Abusive Language Online*, Association for Computational Linguistics, Vancouver, BC, Canada.

Waseem, Z., T. Davidson, D. Warmsley, and I. Weber (2017b), Understanding Abuse: A Typology of Abusive Language Detection Subtasks, in *Proceedings of the First Workshop on Abusive Language Online*, pp. 78–84, Association for Computational Linguistics, Vancouver, BC, Canada.

Weerts, H. J. P., W. van Ipenburg, and M. Pechenizkiy (2019), A Human-Grounded Evaluation of SHAP for Alert Processing, *arXiv:1907.03324 [cs, stat]*, arXiv: 1907.03324.

Whittaker, P. (2015), How one tragic case changed the laws about medical consent for all of us, *New Statesman*.

Wiegreffe, S., and Y. Pinter (2019), Attention is not not Explanation, *arXiv:1908.04626 [cs]*, arXiv: 1908.04626.

Williams, R. J. (1992), Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine learning*, *8*(3-4), 229–256.

Wu, M., M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez (2017), Beyond Sparsity: Tree Regularization of Deep Models for Interpretability, *arXiv:1711.06178 [cs, stat]*, arXiv: 1711.06178.

Wulczyn, E., N. Thain, and L. Dixon (2017), Ex Machina: Personal Attacks Seen at Scale, in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1391–1399, doi: 10.1145/3038912.3052591.