TED: Teaching AI to Explain its Decisions

Noel C. F. Codella,[*] Michael Hind,[*] Karthikeyan Natesan Ramamurthy,[*]
Murray Campbell, Amit Dhurandhar, Kush R. Varshney, Dennis Wei, Aleksandra Mojsilović

IBM Research AI

Abstract

Artificial intelligence systems are being increasingly deployed due to their potential to increase the efficiency, scale, consistency, fairness, and accuracy of decisions. However, as many of these systems are opaque in their operation, there is a growing demand for such systems to provide explanations for their decisions. Conventional approaches to this problem attempt to expose or discover the inner workings of a machine learning model with the hope that the resulting explanations will be meaningful to the consumer. In contrast, this paper suggests a new approach to this problem. It introduces a simple, practical framework, called Teaching Explanations for Decisions (*TED*), that provides meaningful explanations that match the mental model of the consumer. We illustrate the generality and effectiveness of this approach with two different examples, resulting in highly accurate explanations with no loss of prediction accuracy for these two examples.

1 Introduction

Machine learning based systems have proven to be quite effective for producing highly accurate results in several domains. This effectiveness is leading to wider adoption in higher stakes domains, which has the *potential* to lead to more accurate, consistent, and fairer decisions and the resulting societal benefits. However, given the higher stakes of these domains, there is a growing demand that these systems provide explanations for their decisions, so that necessary oversight can occur, and a citizen's due process rights are respected (Goodman and Flaxman 2016; Wachter, Mittelstadt, and Floridi 2017b; Vacca 2018; Campolo, Whittaker, and Crawford 2017; Kim 2017; Doshi-Velez et al. 2017; Wachter, Mittelstadt, and Floridi 2017a; Caruana et al. 2015; Varshney 2016).

The demand for explanation has manifested itself in new regulations that call for automated decision making systems to provide "meaningful information" on the logic used to reach conclusions (Goodman and Flaxman 2016; Wachter, Mittelstadt, and Floridi 2017b; Selbst and Powles 2017). Selbst and Powles (2017) interpret the concept of "meaningful information" as information that should be understandable to the audience (potentially individuals who lack specific expertise), is actionable, and is flexible enough to support various technical approaches.

Unfortunately, the advance in effectiveness of machine-learning techniques has coincided with increased complexity in the inner workings of these techniques. For some techniques, like deep neural networks or large random forests, even experts cannot explain how decisions are reached. Thus, we have a stronger need for explainable AI, just when we have a greater gap in achieving it.

This has sparked a growing research community focused on this problem (Kim et al. 2017; Kim, Varshney, and Weller 2018). Most of this research attempts to explain the inner workings of a machine learning model either directly, indirectly via a simpler proxy model, or by probing the model with related inputs. This paper proposes a different approach that requires a model to jointly produce both a decision as well as an explanation, rather than exposing the inner details of how the model produces a decision. The explanation is not constrained to any particular format and can vary to accommodate user needs.

The main contributions of this work are as follows:

- A description of the challenges in providing meaningful explanations for machine learning systems.
- A new framework, called TED, that enables machine learning algorithms to provide meaningful explanations that match the complexity and domain of consumers.
- A simple instantiation of the framework that demonstrates the generality and simplicity of the approach.
- Two illustrative examples and results that demonstrate the effectiveness of the instantiation in providing meaningful explanations.
- A discussion on several possible extensions and open research opportunities that this framework enables.

The rest of this paper is organized as follows. Section 2 explores the challenges presented by the problem statement of providing explanations for AI decisions. Section 3 discusses related work. Section 4 describes our general approach, TED, and a simple instantiation for providing explanations that are understandable by the consumer and discusses the advantages of the approach. Section 5 presents results from two examples that demonstrate the effectiveness of the simple instantiation. Section 6 discusses future

[*] These authors contributed equally.

directions and open issues for the TED approach. Section 7 draws conclusions.

2 Challenges to Providing AI Explanations

This section explores the challenges in providing meaningful explanations, which provide the motivation for the TED framework.

The concept of an explanation is probably as old as human communication. Intuitively, an explanation is the communication from one person (A) to another (B) that provides justification for an action or decision made by person A. Mathematicians use proofs to formally provide explanations. These are constructed using agreed-upon logic and formalism, so that any person trained in the field can verify if the proof/explanation is valid. Unfortunately, we do not have such formalism for non-mathematical explanations. Even in the judicial system, we utilize nonexpert jurors to determine if a defendant has violated a law, relying on their intuition and experience in weighing (informal) arguments made by prosecution and defense.

Since we do not have a satisfying formal definition for valid human-to-human explanations, developing one for system-to-human explanations is challenging (Kim 2017; Lipton 2016). Motivated by the concept of meaningful information (Goodman and Flaxman 2016; Wachter, Mittelstadt, and Floridi 2017b; Selbst and Powles 2017), we feel that explanations must have the following three characteristics:

Justification: An explanation needs to provide justification for a decision that increases trust in the decision. This often includes some information that can be verified by the consumer.

Complexity Match: The *complexity of the explanation* needs to match the complexity capability of the consumer (Kulesza et al. 2013; Dhurandhar et al. 2017). For example, an explanation in equation form may be appropriate for a statistician, but not for a nontechnical person (Miller, Howe, and Sonenberg 2017).

Domain Match: An explanation needs to be *tailored to the domain*, incorporating the relevant terms of the domain. For example, an explanation for a medical diagnosis needs to use terms relevant to the physician (or patient) who will be consuming the prediction.

There are at least four distinct groups of people who are interested in explanations for an AI system, with varying motivations.

Group 1: End User Decision Makers: These are the people who use the recommendations of an AI system to make a decision, such as physicians, loan officers, managers, judges, social workers, etc. They desire explanations that can build their trust and confidence in the system's recommendations and possibly provide them with additional insight to improve their future decisions and understanding of the phenomenon.

Group 2: Affected Users: These are the people impacted by the recommendations made by an AI system, such as patients, loan applicants, employees, arrested individuals, at-risk children, etc. They desire explanations that can help them understand if they were treated fairly and what factor(s) could be changed to get a different result (Doshi-Velez et al. 2017).

Group 3: Regulatory Bodies: Government agencies, charged to protect the rights of their citizens, want to ensure that decisions are made in a safe and fair manner, and that society is not negatively impacted by the decisions.

Group 4: AI System Builders: Technical individuals (data scientists and developers) who build or deploy an AI system want to know if their system is working as expected, how to diagnose and improve it, and possibly gain insight from its decisions.

Understanding the motivations and expectations behind each group's needs for an explanation will help to ensure a solution that satisfies these expectations. For example, Group 4 is likely to desire a more complex explanation of the system's inner workings to take action. Group 3's needs may be satisfied by showing the overall process, including training data, is fair and free of negative societal impact and they may not be able to consume the same level of complexity as Group 4. Group 1 will have a high need for domain sophistication, but will also have less tolerance for complex explanations. Finally, Group 2 will have the lowest threshold for both complexity and domain information. These are affected users, such as loan applicants, and need to have the reasons for their outcomes such as loan denials explained in a simple manner without industry terms or complex formulas.

In summary, outside of a logical proof, there is no clear definition of a valid explanation; it seems to be subjective to the consumer and circumstances. Furthermore, there is a wide diversity of potential consumers of explanations, with different needs, different levels of sophistication, and different levels of domain knowledge. This seems to make it impossible to produce a single meaningful explanation without any information about the consumer.

3 Related Work

Prior work in providing explanations can be partitioned into three areas:

1. Making existing or enhanced models *interpretable*, i.e. to provide a precise description of how the model determined its decision (e.g., (Ribeiro, Singh, and Guestrin 2016; Montavon, Samek, and Müller 2017; Lundberg and Lee 2017)).
2. Creating a second, simpler-to-understand model, such as a small number of logical expressions, that mostly matches the decisions of the deployed model (e.g., (Bastani, Kim, and Bastani 2018; Caruana et al. 2015)).
3. Work in the natural language processing and computer vision domains that generate rationales/explanations derived from input text (e.g., (Lei, Barzilay, and Jaakkola 2016; Ainur, Choi, and Cardie 2010; Hendricks et al. 2016)).

The first two groups attempt to precisely describe how a machine learning decision was made, which is particularly relevant for AI system builders (Group 4). The insight into the inner workings of a model can be used to improve the AI system and may serve as the seeds for an explanation to a non-AI expert. However, work still remains to determine if these seeds are sufficient to satisfy the needs of the diverse collection of non-AI experts (Groups 1–3). Furthermore, when the underlying features are not human comprehensible, these approaches are inadequate for providing human consumable explanations.

The third group seeks to generate textual explanations with predictions. For text classification, this involves selecting the minimal necessary content from a text body that is sufficient to trigger the classification. For computer vision (Hendricks et al. 2016), this involves utilizing textual captions in training to automatically generate new textual captions of images that are both descriptive as well as discriminative. Although promising, it is not clear how these techniques generalize to other domains and if the explanations will be meaningful to the variety of explanation consumers described in Section 2.

Doshi-Velez et al. (2017) discuss the societal, moral, and legal expectations of AI explanations, provide guidelines for the content of an explanation, and recommend that explanations of AI systems be held to a similar standard as humans. Our approach is compatible with their view. Biran and Cotton (2017) provide an excellent overview and taxonomy of explanations and justifications in machine learning.

Miller (2017) and Miller, Howe, and Sonenberg (2017) argue that explainable AI solutions need to meet the needs of the users, an area that has been well studied in philosophy, psychology, and cognitive science. They provides a brief survey of the most relevant work in these fields to the area of explainable AI. They, along with Doshi-Velez and Kim (2017), call for more rigor in this area.

4 Teaching Explanations

Given the challenges to developing meaningful explanations for the diversity of consumers described in Section 2, we advocate a non-traditional approach. We suggest a high-level framework, with one simple instantiation, that we see as a promising complementary approach to the traditional "inside-out" approach to providing explanations.

To understand the motivation for the TED approach, consider the common situation when a new employee is being trained for their new job, such as a loan approval officer. The supervisor will show the new employee several example situations, such as loan applications, and teach them the correct action: approve or reject, and explain the reason for the action, such as "insufficient salary". Over time, the new employee will be able to make independent decisions on new loan applications and will give explanations based on the explanations they learned from their supervisor. This is analogous to how the TED framework works. We ask the training dataset to teach us, not only how to get to the correct answer (approve or reject), but also to provide the correct explanation, such as "insufficient salary", "too much existing debt", "insufficient job stability", "incomplete application", etc. From this training information, we will generate a model that, for new input, will predict answers and provide explanations based on the explanations it was trained on. Because these explanations are the ones that were provided by the training, they are relevant to the target domain and meet the complexity level of the explanation consumer.

Previous researchers have demonstrated that providing explanations with the training dataset may not add much of a burden to the training time and may improve the overall accuracy of the training data (Zaidan and Eisner 2007; 2008; Zhang, Marshall, and Wallace 2016; McDonnell et al. 2016).

4.1 TED Framework and a Simple Instantiation

The TED framework leverages existing machine learning technology in a straightforward way to generate a classifier that produces explanations along with classifications. To review, a supervised machine learning algorithm takes a training dataset that consists of a series of instances with the following two components:

X, a set of features (feature vector) for the particular entity, such as an image, a paragraph, loan application, etc.

Y, a label/decision/classification for each feature vector, such an image description, a paragraph summary, or a loan-approval decision.

The TED framework requires a third component:

E, an explanation for each decision, Y, which can take any form, such as a number, text string, an image, a video file, etc. Unlike traditional approaches, E does not necessarily need to be expressed in terms of X. It could be some other high-level concept specific to the domain that applies with some domain-specific combination of X, such as "scary image" or "loan information is not trustworthy". Regardless of the format, we represent each unique value of E with an identifier.

The TED framework takes this augmented training set and produces a classifier that predicts both Y and E. There are several ways that this can be accomplished. The instantiation we explore in this work is a simple Cartesian product approach. This approach encodes Y and E into a new classification, called YE, which, along with the feature vector, X, is provided as the training input to any machine learning classification algorithm to produce a classifier that predicts YE's. After the model produced by the classification algorithm makes a prediction, we apply a decoding step to partition a YE prediction into its components, Y and E, to return to the consumer. Figure 1 illustrates the algorithm. The boxes in dashed lines are new TED components that encode Y and E into YE and decode a predicted YE into its individual components, Y and E. The solid boxes represent 1) any machine learning algorithm that takes a normal training dataset: features and labels, and 2) the resulting model produced by this algorithm.

4.2 Example

Let's assume we are training a system to recommend cancer treatments. A typical training set for such a system would

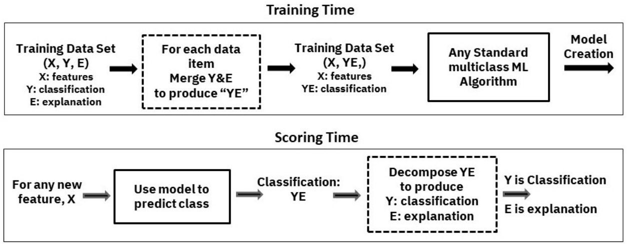

Figure 1: Overview of TED Algorithm

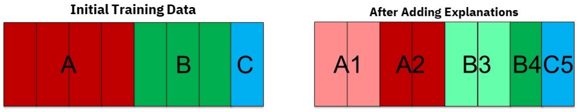

Figure 2: Illustration of Changes to Training Dataset

be of the following form, where P_i is the feature vector representing patient i and T_j, represents various treatment recommendations.

$$(P_1, T_A), (P_2, T_A), (P_3, T_A), (P_4, T_A)$$
$$(P_5, T_B), (P_6, T_B), (P_7, T_B), (P_8, T_C)$$

The TED approach would require adding an additional explanation component to the training dataset as follows:

$$(P_1, T_A, E_1), (P_2, T_A, E_1), (P_3, T_A, E_2), (P_4, T_A, E_2)$$
$$(P_5, T_B, E_3), (P_6, T_B, E_3), (P_7, T_B, E_4), (P_8, T_C, E_5)$$

Each E_i would be an explanation to justify why a feature vector representing a patient would map to a particular treatment. Some treatments could be recommended for multiple reasons/explanations. For example, treatment T_A is recommended for two different reasons, E_1 and E_2, but treatment T_C is only recommended for reason E_5.

Given this augmented training data, the Cartesian product instantiation of the TED framework transforms this triple into a form that any supervised machine learning algorithm can use, namely (feature, class) by combining the second and third components into a unique new class as follows:

$$(P_1, T_AE_1), (P_2, T_AE_1), (P_3, T_AE_2), (P_4, T_AE_2)$$
$$(P_5, T_BE_3), (P_6, T_BE_3), (P_7, T_BE_4), (P_8, T_CE_5)$$

Figure 2 shows how the training dataset would change using the TED approach for the above example. The left picture illustrates how the original 8 training instances in the example are mapped into the 3 classes. The right picture shows how the training data is changed, with explanations added. Namely, Class A was decomposed to Classes $A1$ and $A2$. Class B was transformed into Classes $B3$ and $B4$ and Class C became $C5$.

As Figure 2 illustrates, adding explanations to training data implicitly creates a 2-level hierarchy in that the transformed classes are members of the original classes, e.g., Classes $A1$ and $A2$ are a decomposition of the original Class A. This hierarchical property could be exploited by employing hierarchical classification algorithms when training to improve accuracy.

4.3 Advantages

Although this approach is simple, there are several nonobvious advantages that are particularly important in addressing the requirements of explainable AI for groups 1, 2, and 3 discussed in Section 2.

Complexity/Domain Match: Explanations provided by the algorithm are guaranteed to match the complexity and mental model of the domain, given that they are created by the domain expert who is training the system.

Dealing with Incomprehensible Features: Since the explanation format can be of any type, they are not limited to being a function of the input features, which is useful when the features are not comprehensible.

Accuracy: Explanations will be accurate if the training data explanations are accurate and representative of production data.

Generality: This approach is independent of the machine learning classification algorithm; it can work with any supervised classification algorithm, including neural networks, making this technique widely deployable.

Preserves Intellectual Property: There is no need to expose details of machine learning algorithm to the consumer. Thus, proprietary technology can remain protected by their owners.

Easy to incorporate: The Cartesian product approach does not require a change to the current machine learning algorithm, just the addition of pre- and post-processing components: encoder and decoder. Thus, an enterprise does not need to adopt a new machine learning algorithm, just to get explanations.

Educates Consumer: The process of providing good training explanations will help properly set expectations for what kind of explanations the system can realistically provide. For example, it is probably easier to explain in the training data why a particular loan application is denied than to explain why a particular photo is a cat. Setting customer expectations correctly for what AI systems can (currently) do is important to their satisfaction with the system.

Improved Auditability: After creating a TED dataset, the domain expert will have enumerated all possible explanations for a decision. (The TED system does not create any new explanations.) This enumeration can be useful for the consumer's auditability, i.e., to answer questions such as "What are the reasons why you will deny a loan?" or "What are the situations in which you will prescribe medical treatment X?"

May Reduce Bias: Providing explanations will increase the likelihood of detecting bias in the training data because 1) a biased decision will likely be harder for the explanation producer to justify, and 2) one would expect that training instances with the same explanations cluster close to each other in the feature space. Anomalies from this property could signal a bias or a need for more training data.

5 Evaluation

To evaluate the ideas presented in this work, we focus on two fundamental questions:

1. How useful are the explanations produced by the TED ap-

proach?

2. How is the prediction accuracy impacted by incorporating explanations into the training dataset?

Since the TED framework has many instantiations, can be incorporated into many kinds of learning algorithms, tested against many datasets, and used in many different situations, a definitive answer to these questions is beyond the scope of this paper. Instead we try to address these two questions using the simple Cartesian product instantiation with two different machine learning algorithms (neural nets and random forest), on two use cases to show that there is justification for further study of this approach.

Determining if any approach provides useful explanations is a challenge and no consensus metric has yet to emerge (Doshi-Velez et al. 2017). However, since the TED approach requires explanations be provided for the target dataset (training and testing), one can evaluate the accuracy of a model's explanation (E) in a similar way that one evaluates the accuracy of a predicted label (Y).

The TED approach requires a training set that contains explanations. Since such datasets are not yet readily available, we evaluate the approach on two synthetic datasets described below: tic-tac-toe and loan repayment.

5.1 Tic-Tac-Toe

The tic-tac-toe example tries to predict the best move given a particular board configuration. A tic-tac-toe board is represented by two 3×3 binary feature planes, indicating the presence of X and O, respectively. An additional binary feature indicates the side to move, resulting in a total of 19 binary input features. Each legal non-terminal board position (4,520) is labeled with a preferred move, along with the reason the move is preferred. The labeling is based on a simple set of rules that are executed sequentially:[1]

1. If a winning move is available, completing three in a row for the side to move, choose that move with reason *Win*
2. If a blocking move is available, preventing the opponent from completing three in a row on their next turn, choose that move with reason *Block*
3. If a threatening move is available, creating two in a row with an empty third square in the row, choose that move with reason *Threat*
4. Otherwise, choose an empty square, preferring center over corners over middles, with reason *Empty*

Two versions of the dataset were created, one with only the preferred move (represented as a 3×3 plane), the second with the preferred move and explanation (represented as a $3 \times 3 \times 4$ stack of planes). A simple neural network classifier was built on each of these datasets, with one hidden layer of 200 units using ReLU and a softmax over the 9 (or 36) outputs. We use a 90%/10% split of the legal non-terminal board positions for the training/testing datasets. This classifier obtained an accuracy of 96.5% on the baseline move-only prediction task, i.e., when trained with just X (the 19 features) and Y it was highly accurate.

Table 1: Accuracy for predicting Y and E in Tic-Tac-Toe and Loan Repayment

Training Input	Accuracy (%)			
	Tic-Tac-Toe		Loan Repayment	
	Y	E	Y	E
X, Y	96.5	NA	99.2 (0.2)	NA
X, Y, and E	97.4	96.3	99.6 (0.1)	99.4 (0.1)

To answer the first question, does the approach provide useful explanations, we calculated the accuracy of the predicted explanation. Although there are only 4 rules, each rule applies to 9 different preferred moves, resulting in 36 possible explanations. Our classifier was able to generate the correct explanation 96.3% of the time, i.e., very rarely did it get the correct move and not the correct rule.

The second question asks how the accuracy of the classifier is impacted by the addition of E's in the training dataset. Given the increase in number of classes, one might expect the accuracy to decrease. However, for this example, the accuracy of predicting the preferred move actually increases to 97.4%. This illustrates that the approach works well in this domain; it is possible to provide accurate explanations without impacting the Y prediction accuracy. Table 1 summarizes the results for both examples.

5.2 Loan Repayment

The second example is closer to an industry use case and is based on the FICO Explainable Machine Learning Challenge dataset (FICO 2018). The dataset contains around 10,000 applications for Home Equity Line of Credit (HELOC), with the binary Y label indicating payment performance (any 90-day or longer delinquent payments) over 2 years.

Since the dataset does not come with explanations (E),[2] we generated them by training a rule set on the training data, resulting in the following two 3-literal rules for the "good" class $Y = 1$ (see (FICO 2018) for a data dictionary):

1. NumSatisfactoryTrades $\geq$ 23 AND
   ExternalRiskEstimate $\geq$ 70 AND
   NetFractionRevolvingBurden $\leq$ 63;
2. NumSatisfactoryTrades $\leq$ 22 AND
   ExternalRiskEstimate $\geq$ 76 AND
   NetFractionRevolvingBurden $\leq$ 78.

These two rules, from researchers at IBM Research, predict Y with 72% accuracy and were the winning entry to the challenge. Since the TED approach requires 100% consistency between explanations and labels, we modified the Y labels in instances where they disagree with the rules. We then assigned the explanation E to one of 8 values: 2 for the good class, corresponding to which of the two rules is satisfied (they are mutually exclusive), and 6 for delinquent, corresponding first to which of the rules should apply based

[1] These rules do not guarantee optimal play.

[2] The challenge asks participants to provide explanations along with predictions, which will be judged by the organizers.

on NumSatisfactoryTrades, and then to which of the remaining conditions (ExternalRiskEstimate, NetFractionRevolvingBurden, or both) are violated.

We trained a Random Forest classifier (100 trees, minimum 5 samples per leaf) on first the dataset with just X and (modified) Y and then on the enhanced dataset with E added. The accuracy of the baseline classifier (predicting binary label Y) was 99.2%. The accuracy of TED in predicting explanations E was 99.4%, despite the larger class cardinality of 8. In this example, Y predictions can be derived from E predictions through the mapping mentioned above, and doing so resulted in an improved Y accuracy of 99.6%. While these accuracies may be artificially high due to the data generation method, they do show two things as in Section 5.1: (1) To the extent that user explanations follow simple logic, very high explanation accuracy can be achieved; (2) Accuracy in predicting Y not only does not suffer but actually improves. The second result has been observed by other researchers who have suggested adding "rationales" to improve classifier performance, but not for explainability (Sun and DeJong 2005; Zaidan and Eisner 2007; 2008; Zhang, Marshall, and Wallace 2016; McDonnell et al. 2016; Donahue and Grauman 2011; Duan et al. 2012; Peng et al. 2016).

6 Extensions and Open Questions

The TED framework assumes a training dataset with explanations and uses it to train a classifier that can predict Y and E. This work described a simple way to do this, by taking the Cartesian product of Y and E and using any suitable machine learning algorithm to train a classifier. Another instantiation would be to bring together the labels and explanations in a multitask setting. Yet another option is to learn feature embeddings using labels and explanation similarities in a joint and aligned way to permit neighbor-based explanation prediction.

Under the Cartesian product approach, adding explanations to a dataset increases the number of classes that the classification algorithm will need to handle. This could stress the algorithm's effectiveness or training time performance, although we did not observe this in our two examples. However, techniques from the "extreme classification" community (Extreme 2017) could be applicable.

Although the flexibility of allowing any format for an explanation, provided the set of explanations can be enumerated, is quite general, it could encourage a large number of explanations that differ in only unintended ways, such as "insufficient salary" vs. "salary too low". Providing more structure via a domain-specific language (DSL) or good tooling could be useful. If free text is used, we could leverage word embeddings to provide some structure and to help reason about similar explanations.

As there are many ways to explain the same phenomenon, it may be useful to explore having more than one version of the same base explanation for different levels of consumer sophistication. Applications already do this for multilingual support, but in this case it would be multiple levels of sophistication in the same language for, say, a first time borrower vs. a loan officer or regulator. This would be a postprocessing step once the explanation is predicted by the classifier.

Providing explanations for the full training set is ideal, but may not be realistic. Although it may be easy to add explanations *while* creating the training dataset, it may be more challenging to add explanations after a dataset has been created because the creator may not available or may not remember the justification for a label. One possibility is to use an external knowledge source to generate explanations, such as WebMD in a medical domain. Another possibility is to request explanations on a subset of the training data and apply ideas from few-shot learning (Goodfellow, Bengio, and Courville 2016) to learn the rest of the training dataset explanations. Another option is to use active learning to guide the user where to add explanations. One approach may be to first ask the user to enumerate the classes and explanations and then to provide training data (X) for each class/explanation until the algorithm achieves appropriate confidence. At a minimum one could investigate how the performance of the explanatory system changes as more training explanations are provided. Combinations of the above may be fruitful.

7 Conclusions

This paper introduces a new paradigm for providing explanations for machine learning model decisions. Unlike existing methods, it does not attempt to probe the reasoning process of a model. Instead, it seeks to replicate the reasoning process of a human domain user. The two paradigms share the objective to produce a reasoned explanation, but the model introspection approach is more suited to AI system builders who work with models directly, whereas the teaching explanations paradigm more directly addresses domain users. Indeed, the European Union GDPR guidelines say: "The controller should find simple ways to tell the data subject about the rationale behind, or the criteria relied on in reaching the decision without necessarily always attempting a complex explanation of the algorithms used or disclosure of the full algorithm."

Work in social and behavioral science (Lombrozo 2007; Miller, Howe, and Sonenberg 2017; Miller 2017) has found that people prefer explanations that are simpler, more general, and coherent, even over more likely ones. Miller writes that in the context of Explainable AI: "Giving simpler explanations that increase the likelihood that the observer both understands and accepts the explanation may be more useful to establish trust (Miller, Howe, and Sonenberg 2017)."

Our two examples illustrate promise for this approach. They both showed highly accurate explanations and no loss in prediction accuracy. We hope this work will inspire other researchers to further enrich this paradigm.

References

Ainur, Y.; Choi, Y.; and Cardie, C. 2010. Automatically generating annotator rationales to improve sentiment classification. In *Proceedings of the ACL 2010 Conference Short Papers*, 336–341.

Bastani, O.; Kim, C.; and Bastani, H. 2018. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*.

Biran, O., and Cotton, C. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*.

Campolo, A.; Whittaker, M. S. M.; and Crawford, K. 2017. 2017 annual report. Technical report, AI NOW.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 1721–1730.

Dhurandhar, A.; Iyengar, V.; Luss, R.; and Shanmugam, K. 2017. A formal framework to characterize interpretability of procedures. In *Proc. ICML Workshop Human Interp. Mach. Learn.*, 1–7.

Donahue, J., and Grauman, K. 2011. Annotator rationales for visual recognition. In *ICCV*.

Doshi-Velez, F., and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. In *https://arxiv.org/abs/1702.08608v2*.

Doshi-Velez, F.; Kortz, M.; Budish, R.; Bavitz, C.; Gershman, S.; O'Brien, D.; Schieber, S.; Waldo, J.; Weinberger, D.; and Wood, A. 2017. Accountability of AI under the law: The role of explanation. *CoRR* abs/1711.01134.

Duan, K.; Parikh, D.; Crandall, D.; and Grauman, K. 2012. Discovering localized attributes for fine-grained recognition. In *CVPR*.

2017. Extreme classification: The nips workshop on multi-class and multi-label learning in extremely large label spaces.

FICO. 2018. Explainable machine learning challenge.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodman, B., and Flaxman, S. 2016. EU regulations on algorithmic decision-making and a 'right to explanation'. In *Proc. ICML Workshop Human Interp. Mach. Learn.*, 26–30.

Hendricks, L. A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; and Darrell, T. 2016. Generating visual explanations. In *European Conference on Computer Vision*.

Kim, B.; Malioutov, D. M.; Varshney, K. R.; and Weller, A., eds. 2017. *2017 ICML Workshop on Human Interpretability in Machine Learning*.

Kim, B.; Varshney, K. R.; and Weller, A., eds. 2018. *2018 Workshop on Human Interpretability in Machine Learning*.

Kim, B. 2017. Tutorial on interpretable machine learning.

Kulesza, T.; Stumpf, S.; Burnett, M.; Yang, S.; Kwan, I.; and Wong, W.-K. 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput.*, 3–10.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. In *EMNLP*.

Lipton, Z. C. 2016. The mythos of model interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*.

Lombrozo, T. 2007. Simplicity and probability in causal explanation. *Cognitive Psychol.* 55(3):232–257.

Lundberg, S., and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances of Neural Inf. Proc. Systems*.

McDonnell, T.; Lease, M.; Kutlu, M.; and Elsayed, T. 2016. Why is that relevant? collecting annotator rationales for relevance judgments. In *Proc. AAAI Conf. Human Comput. Crowdsourc.*

Miller, T.; Howe, P.; and Sonenberg, L. 2017. Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences. In *Proc. IJCAI Workshop Explainable Artif. Intell.*

Miller, T. 2017. Explanation in artificial intelligence: Insights from the social sciences. *arXiv preprint arXiv:1706.07269*.

Montavon, G.; Samek, W.; and Müller, K.-R. 2017. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*.

Peng, P.; Tian, Y.; Xiang, T.; Wang, Y.; and Huang, T. 2016. Joint learning of semantic and latent attributes. In *ECCV 2016, Lecture Notes in Computer Science*, volume 9908.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 1135–1144.

Selbst, A. D., and Powles, J. 2017. Meaningful information and the right to explanation. *Int. Data Privacy Law* 7(4):233–242.

Sun, Q., and DeJong, G. 2005. Explanation-augmented svm: an approach to incorporating domain knowledge into svm learning. In *22nd International Conference on Machine Learning*.

Vacca, J. 2018. A local law in relation to automated decision systems used by agencies. Technical report, The New York City Council.

Varshney, K. R. 2016. Engineering safety in machine learning. In *Information Theory and Applications Workshop*.

Wachter, S.; Mittelstadt, B.; and Floridi, L. 2017a. Transparent, explainable, and accountable AI for robotics. *Science Robotics* 2.

Wachter, S.; Mittelstadt, B.; and Floridi, L. 2017b. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *Int. Data Privacy Law* 7(2):76–99.

Zaidan, O. F., and Eisner, J. 2007. Using 'annotator rationales' to improve machine learning for text categorization. In *In NAACL-HLT*, 260–267.

Zaidan, O. F., and Eisner, J. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of EMNLP 2008*, 31–40.

Zhang, Y.; Marshall, I. J.; and Wallace, B. C. 2016. Rationale-augmented convolutional neural networks for text classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.