LIMEADE: A General Framework for Explanation-Based Human Tuning of Opaque Machine Learners

Benjamin Charles Germain Lee,1,2* Doug Downey,1,3 Kyle Lo,1 Daniel S. Weld1,2
1Allen Institute for Artificial Intelligence
2The University of Washington
3Northwestern University

Abstract

Research in human-centered AI has shown the benefits of systems that can explain their predictions. Methods that allow humans to tune a model in response to the explanations are similarly useful. While both capabilities are well-developed for transparent learning models (e.g., linear models and GA^2Ms), and recent techniques (e.g., LIME and SHAP) can generate explanations for opaque models, no method for tuning opaque models in response to explanations has been user-tested to date. This paper introduces LIMEADE, a general framework for tuning an arbitrary machine learning model based on an explanation of the model’s prediction. We demonstrate the generality of our approach with two case studies. First, we successfully utilize LIMEADE for the human tuning of opaque image classifiers. Second, we apply our framework to a neural recommender system for scientific papers on a public website and report on a user study showing that our framework leads to significantly higher perceived user control, trust, and satisfaction. Analyzing 300 user logs from our publicly-deployed website, we uncover a tradeoff between canonical greedy explanations and diverse explanations that better facilitate human tuning.

Introduction

Guidelines for human-AI interaction dictate that recommender systems, and machine learning (ML) systems more generally, should be able to explain their predictions and accept corrections (Amershi et al. 2019). Both explanation and tuning methods exist for transparent models, such as linear classifiers or generalized additive models (GA^2Ms) (Caruana et al. 2015), and their benefits for transparent recommenders have been demonstrated (Bostandjiev, O’Donovan, and Höllerer 2012; Kulesza et al. 2012). But since opaque models, such as boosted decision forests and deep neural networks, often provide higher performance, many researchers have developed methods for generating explanations of an opaque ML model — typically creating a transparent approximation to the opaque model, called an explanatory model (Guidotti et al. 2018). While these explanatory models have seen significant adoption, the problem of interpreting human feedback has also received attention in recent years (Ross, Hughes, and Doshi-Velez 2017; Liu and...

*Work performed during internship at the Allen Institute for Artificial Intelligence and Ph.D. at the University of Washington.
Figure 1: An interpretable model, such as a GA²M (left), is by definition explainable and tunable (Caruana et al. 2015). With an opaque model (right), methods such as LIME and SHAP enable the user to receive an approximate explanation, often using a new vocabulary (e.g., superpixels instead of pixels). LIMEADE allows users to provide feedback — using features of the explanatory model — and then modifies the original, opaque model by retraining.

Previous Work

Explainability

Developing methods for understanding the reasons behind a model’s predictions is an essential part of machine learning. Models such as linear models and decision trees benefit from being intrinsically interpretable but are limited in performance on many tasks. In contrast, opaque models such as neural networks are ubiquitous due to high performance but present challenges precisely because they are difficult to understand. Consequently, many approaches to explaining opaque models have been proposed (Guidotti et al. 2018). On the other hand, feature attribution methods such as LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017) have been developed in order to provide post-hoc explanations by creating an approximate explanatory model of any opaque model. However, these methods provide no affordances for tuning the opaque model in response to an explanation, which is especially important in the context of human-centered AI.

Enabling Humans to Tune Machine Learners

Research from interactive machine learning has shown the benefits of enabling humans to tune learning models, including recommender systems and beyond (Amershi, Fogarty, and Weld 2012; Simard et al. 2017). For example, Lou et al. (2012), Lou et al. (2013), and Caruana et al. (2015) have demonstrated the value of GAMs and GA²Ms, which can be directly tuned by humans via the alteration of shape functions. Likewise, Kulesza et al. (2015) have shown the power of explanatory debugging of models. However, this research has focused on transparent, interpretable models, where the models can be tuned directly (Weld and Bansal 2019). LIMEADE extends the paradigm of interactive machine learning and tuning to opaque models. While other work has explored the extension of this paradigm to other classes of opaque models (Dasgupta et al. 2019; Ross, Hughes, and Doshi-Velez 2017; Liu and Avcı 2019; Kieger et al. 2020; Schramowski et al. 2020; Bau et al. 2020; Mu and Andreas 2020), none of these works have studied the tuning of explanations via user experiments, an essential form of evaluation in human-AI interaction research (Amershi et al. 2019).
Labeling Features and Creating Pseudo-examples

One approach to semi-supervised learning is to train a machine learner on labeled examples, as well as labeled features via the construction of pseudo-examples (Wu and Srihari 2004; Druck, Mann, and McCallum 2008; Raghavan and Allan 2007; Godbole et al. 2004; Liu et al. 2004). In the text classification setting, this often takes the form of labeling n-gram features, which are then used to construct pseudo-documents, e.g. containing just the labeled n-gram itself, labeled according to the feature’s assigned label. LIMEADE extends this semi-supervised approach to feature labeling for opaque models by translating labels on features in the interpretable space to pseudo-examples in the featurization of the opaque model.

Explaining & Tuning Image Classifiers

For our first case study, we evaluate LIMEADE in the context of tuning an image classifier, a setting frequently adopted for investigating explainability and tunability. For example, the papers introducing LIME and SHAP demonstrate the utility of their frameworks by explaining image classifiers (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017). Likewise, a significant fraction of existing work on tuning opaque models has adopted the setting of debugging image classifiers (Dasgupta et al. 2019; Rieger et al. 2020; Ross, Hughes, and Doshi-Velez 2017; Bau et al. 2020; Schramowski et al. 2020; Mu and Andreas 2020). Of this work, (Rieger et al. 2020; Doshi-Velez and Kim 2017; Bau et al. 2020; Schramowski et al. 2020; Mu and Andreas 2020) all focus on updates to deep neural models, while (Doshi-Velez and Kim 2017) studies differentiable models whose gradients can be accessed; only (Dasgupta et al. 2019) studies black-box learners in the most general case, where the update only requires being able to query the opaque model with examples. We, too, consider the setting of image classification to show the generalizability of LIMEADE, which can be used to update all opaque models satisfying the conditions enumerated in the next section of the paper.

Explaining & Tuning Recommender Systems

Our second case study focuses on an academic paper recommendation system. Recommender systems are a highly-studied domain for explainability and tunability due to the feedback loop and interactivity essential to the task of recommendation (Ahn, Brusilovsky, and Han 2015; Brusilovsky et al. 2020; He, Parra, and Verbert 2016; Loepp, Babu, and Ziegler 2016; Pu, Chen, and Hu 2011; Tintarev and Masthoff 2011; Tsai and Brusilovsky 2018; 2019; Wærn 2004; Zhang and Chen 2018). Some recommendation systems enable the human to tune the system via affordances other than rating content (Ahn et al. 2007; Bakalov et al. 2013; Bostandjiev, O’Donovan, and Hölzer 2012; Bostandjiev, O’Donovan, and Höllerer 2013; Bruns et al. 2015; Gretarsdottir et al. 2010; Harper et al. 2015; He, Parra, and Verbert 2016; Jin, Tintarev, and Verbert 2018; Kangasrääsiö, Glowacka, and Kaski 2015; Knijnenburg, Reijmer, and Willemse 2011; Kulesza et al. 2012; O’Donovan et al. 2008; Parra and Brusilovsky 2015; Tsai and Brusilovsky 2020; Schaffer, Höllerer, and O’Donovan 2015; Vig, Sen, and Riedl 2012). The combined affordances of tunability and explainability can lead to a higher degree of user satisfaction (Herlocker, Konstan, and Riedl 2000); more trust in and perceived control of the system (Cramer et al. 2008; Herlocker, Konstan, and Riedl 2000; Lu and Chen 2006; Verbert et al. 2013); and better mental models, without significantly increasing the cognitive load (Kulesza et al. 2015; Kulesza et al. 2015; Rosenthal and Dey 2010). However, all of this work either relies on interpretable recommenders or interprets tunability in an algorithmic-specific fashion that is not extensible to an arbitrary opaque model. With Semantic Sanity, we demonstrate with user testing the benefits of LIMEADE’s affordance for tuning an arbitrary opaque model.

Of particular interest are the results from Ahn et al. (Ahn et al. 2007) and Wærn (Wærn 2004), who demonstrated that tunability for search and recommendation tasks can negatively impact feed quality when it takes the form of adding or removing terms from the featurization. However, these affordances do not allow the human to ask for more or less of a term, just simply to add or remove it. Notably, (Ahn et al. 2007) found that removing terms had about 4 times the negative impact of adding terms, likely due to the fact that removing salient keywords provided by the system has a significant impact on feed rankings. In Semantic Sanity, we provide the affordance of asking for more or less of a term.

The subarea of academic paper recommendation is especially important because the overwhelming influx of new scientific publications poses a daily challenge for researchers (Bhagavatula et al. 2018; Ekstrand et al. 2010; He et al. 2010; Kanakia et al. 2019; Sinha et al. 2015). However, based on Beel et al. (2016)’s survey of 185 publications on academic paper recommendation, only a few systems explain why papers have been recommended or respond to user feedback other than liking/disliking specific papers, and all such systems rely on interpretable recommenders (Bakalov et al. 2013; Kangasrääsiö, Glowacka, and Kaski 2015; Bruns et al. 2015; Verbert et al. 2013; Parra and Brusilovsky 2015). The ability to explain and tune higher-performance paper recommenders, therefore, would fill an important void.

LIMEADE: Human Tuning of Opaque Models

With LIMEADE, we assume that the human would like to tune an opaque machine learning model such as a neural recommender system. By opaque, we mean that the model architecture may be completely unknown, or (if known), it may have too many parameters and nonlinearities for a human to understand. However, we assume that the model’s inputs and outputs are available and that the model can be retrained on new examples. We work in a semi-supervised learning setting, in which the goal is to learn a hypothesis that maps an s-dimensional real-valued input vector to a real-valued output score in $[-1, 1]$ (a classification setting is also possible here, as explored in the next section). We are given a set $X_L$ of labeled training examples $(x,y,w)$, where $x \in \mathbb{R}^s$, $y$ is the value to be learned, and $w$ is the
weight assigned to the example when training. Additionally, we optionally have a large, dense pool $\mathcal{X}_U$ of unlabelled examples ($x$). Our explainable machine learning problem setting closely follows that of previous work in explainable ML (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017). We assume that each instance $x$ can be represented as a binary-valued vector $x'$ that lies in an interpretable space. For the example of paper recommendation, $x$ is an opaque document embedding output by a large neural network, whereas $x'$ is an interpretable vector of term occurrence statistics. In the text domain, the dimensions of $x'$ would correspond to interpretable features such as TF-IDF values for n-grams.

The model $g$ can be any interpretable model, such as a decision tree or linear model, produced using LIME or a comparable method. We refer to the method that produces $g$ as EXPLAIN($f, x, h'$). In our experiments, we use a linear explanatory model, so $g(x') = w_0 + \sum_i w_i x'_i$, and the explanation surfaced for $g(x')$ consists of high-impact terms in the model, i.e., those with high values for the product $w_i x'_i$.

Algorithm 1 Tuning an opaque model using LIMEADE. Given a set of required inputs, LIMEADE solicits human tuning based on an explanation of a classified instance and retrains the opaque model accordingly. EXPLAIN is a function that generates an explanation for a given model and instance.

Inputs:

- $\mathcal{X}_L, \mathcal{X}_U$: sets of labeled and unlabelled instances
- $f_t: \mathbb{R}^n \rightarrow [-1, 1]$: opaque classifier, version at time $t$
- $x \in \mathbb{R}^n, x' \in \{0, 1\}^s$: instance & instance in interpret. rep.
- $h': \mathbb{R}^n \rightarrow \{0, 1\}^s$: mapping s.t. $x' = h'(x)$
- $\pi_x: \{0, 1\}^s \rightarrow \mathcal{X}_L$: weighting based on distance
- $k \in \mathbb{N}$: number of pseudo-examples

1: $g_t = \text{EXPLAIN}(f_t, x, h')$ // obtain explanatory model
2: // display key features of $g_t(x')$ to user, who then selects one feature (indexed $j$) as $+$ or $-$ indicator of positive instance
3: receive $a \in \{-1, 1\}$ and $j \in \{1, \ldots, s\}$
4: // select $k$ instances, label them using action $a$, and weight according to distance from $x'$
5: $\mathcal{N}_x \leftarrow \{\}$
6: for $i = 1, \ldots, k$ do
7: $\hat{x} = \text{GETINSTANCE}(x, x', \mathcal{X}_U)$
8: if $h'(\hat{x})[j] = 1$ then
9: $\mathcal{N}_x \leftarrow \mathcal{N}_x \cup \{(\hat{x}, a, \pi_x(h'(\hat{x})))\}$
10: end if
11: end for
12: $f_{t+1} \leftarrow \text{RETRAIN}(\mathcal{X}_L, f_t)$
13: return $f_{t+1}$

Algorithm 1 details LIMEADE’s approach to model tuning, and Figure 2 illustrates a concrete example of applying LIMEADE. Given an instance of interest, $x$, we obtain an explanation $g(x')$ of the model’s output $f(x)$ using EXPLAIN($f, x, h'$). The human can then provide a label on a feature of $x'$. Informally, a positive label on feature $j$ of $x'$ represents the human’s assessment that examples $z'$ near $x'$ should tend to be positive when $z'[j] = 1$. For example, a user of our paper recommendation system might give a positive label to the term “BERT” in a natural language processing paper to indicate interest in papers about the technique.

LIMEADE uses the human’s action to improve the opaque model $f$ by creating a set of $k$ training pseudo-examples with repeated calls to GETINSTANCE($x, x', \mathcal{X}_U$). We experiment with two implementations of GETINSTANCE: sampling and generative. Sampling from the unlabeled pool is effective when the unlabeled pool is relatively dense, meaning one can acquire many examples with interpretable features similar to those of $x'$. Generative approaches can be helpful when data are less dense. For example, with images, LIMEADE could create synthetic pseudo-examples by greying out random subsets of the superpixels in the input image, essentially reversing LIME’s process for generating the explanatory model, $g$.

LIMEADE only retains the pseudo-examples that contain the acted-upon feature $j$, i.e., those $\hat{x}$ for which $h'(\hat{x})[j] = 1$. LIMEADE then assigns a value to each pseudo-example according to the user action: $+1$ if the user assigned a positive feature label, and $-1$ otherwise.

LIMEADE assigns each pseudo-example a weight based on its proximity to $x'$, with examples more similar to $x'$ given higher weight. The reasons to weight local examples more highly are twofold: the explanatory method may only be locally correct (Ribeiro, Singh, and Guestrin 2016), and the human actions may only be locally applicable. For example, the positive label on “BERT” discussed earlier is helpful within the local scope of natural language processing papers, but could become misleading if applied globally—in biology papers for example, the term “BERT” often refers to a different meaning (the “BERT gene”). After selecting and weighting the pseudo-examples, LIMEADE can optionally condense the selections (e.g., collapsing the examples into a single centroid for efficiency, as we do in our experiments). Finally, LIMEADE adds the resulting pseudo-examples to the labeled training set $\mathcal{X}_L$ and calls RETRAIN to train the classifier $f$ on the new data set.

Applying LIMEADE to Different Domains & Tasks

We note that LIMEADE is a general framework for the human tuning of an opaque model, whether a rank-oriented recommender or discrete classifier. LIMEADE is applicable to an interactive machine learning task whenever the following requirements are met:

1. A base (opaque) model. We require the ability to retrain the model and access the original training data.

---

1We measure proximity in the interpretable space, but it is equally possible to measure in the original space instead.
2. A transparent (Doshi-Velez and Kim 2017) explanatory model (such as one produced by LIME). This explanatory model can be either a local or global approximation (Ribeiro, Singh, and Guestrin 2016).

3. The ability to sample or generate instances with featureizations in both the base and explanatory models.

4. The ability to receive a human’s feedback on an explanatory model feature, indicating how the presence of this feature should influence a prediction.

5. A distance metric between instances.

6. The ability to detect for a given instance the presence or absence of each explanatory model feature that received feedback in Step 4 of Algorithm 1.

In the following two case studies, we describe successful instantiations of LIMEADE in both the image domain and text domain.

Case Study 1: LIMEADE for Image Classification

To demonstrate how the LIMEADE framework applies to the computer vision domain, we use it to enable end-user updates to a variety of deep neural image classifiers, e.g. a skateboard detector. In Figure 3, we illustrate an example of how LIMEADE can be utilized in this context for tuning.

Experimental Setup

To see if LIMEADE was general enough to support tuning in the image domain, we evaluated on eight different binary image classifiers, each comprising a logistic regression model trained on pre-computed image embeddings.

As a base dataset, we utilized 13,000 images from the COCO dataset (Lin et al. 2014). We created superpixel features using the same segmentation algorithm (Pedregosa et al. 2011) used by LIME. We generated embeddings for all images and superpixels by retrieving their representations from the penultimate layer of a ResNet-50 backbone pre-trained on ImageNet (He et al. 2015; Deng et al. 2009).

For our experiment, we considered binary classifiers for eight different COCO classes: carrot, giraffe, snowboard, fork, bed, skateboard, fire hydrant, & potted plant (these classes were chosen because none appear as classes in ImageNet-1000, thus preventing full semantic alignment of the pre-computed embeddings with our classification tasks). For each class, we evaluated the performance of a LIMEADE update relative to two baseline updates for 50 randomized initial configurations. For each configuration, we randomly constructed an initial training set of one positive and one negative instance, as well as a balanced, held-out set for evaluating performance. We evaluated the two-shot accuracy of a logistic regression model on this training set and then performed the following updates with both a randomly-drawn positive instance and a randomly-drawn negative instance simultaneously to preserve class balance:

1. **Baseline 1**: We updated the model by adding the positive and negative instances to the training data and retraining.

2. **Baseline 2**: Unlike LIMEADE, Baseline 1 does not take advantage of the unlabeled example pool. To isolate
Figure 3: a) Suppose an opaque classifier incorrectly classifies an image of a skier as a positive example of skateboarding. Suppose further that LIME returns an explanation showing a superpixel containing the skier’s helmet as a positive indicator of the skateboarding class. b) While a helmet is an appropriate positive indicator for the skateboarding class, the user may wish to indicate that another superpixel, containing skis and ski poles, is a negative indicator. Given this user feedback, LIMEADE tunes the opaque model by retrieving unlabeled images containing superpixels most similar to this ski superpixel. These images are then added to the training data — with negative labels — and the model is retrained, completing the LIMEADE update.

In Table 1, we report the net changes in classifier accuracy when making updates with LIMEADE and both baselines. Paired t-tests of LIMEADE against Baseline 1 and Baseline 2 yield p-values of 0.0851 and 0.106, respectively. While these results are not statistically significant, they are promising, suggesting that the LIMEADE update meets or outperforms both baselines.

### Results in the Image Domain

In Table 1, we report the net changes in classifier accuracy when making updates with LIMEADE and both baselines. Paired t-tests of LIMEADE against Baseline 1 and Baseline 2 yield p-values of 0.0851 and 0.106, respectively. While these results are not statistically significant, they are promising, suggesting that the LIMEADE update meets or outperforms both baselines.

### LIMEADE Updates Perform Better when LIME Explanations are Higher Quality

Because a LIMEADE update requires an explanation of an opaque model’s prediction in order for a human to provide feedback, we investigated whether LIMEADE’s performance is correlated with the quality of a LIME explanation.

To evaluate the quality of a LIME explanation, we considered the LIME explanations in the positive case for all 400 iterations of our experiment. In particular, we evaluated...
the change in the opaque classifier’s prediction probability when greying out the superpixel that LIME determined to be most positively-influencing the opaque model’s prediction. A greater drop in prediction probability indicates that the superpixel identified by LIME has a greater influence over the opaque classifier’s prediction. Thus, this drop in prediction probability is a proxy for LIME’s explanation quality.

We evaluated the correlation between LIMEADE’s performance relative to Baseline 2 as a function of this proxy. We find a positive correlation (slope = 0.053). This indicates that LIMEADE’s performance relative to the baseline is positively correlated with LIME’s explanation quality.

### Case Study 2: LIMEADE for Paper Recommendation

To test LIMEADE, we wanted human users who were authentically motivated to understand and improve an ML classifier. For our second case study, we built Semantic Sanity, a computer science (CS) research-paper recommender system based on Andrej Karpathy’s arXiv Sanity Preserver (2015). Deployed as a publicly-available platform, Semantic Sanity enables users to curate feeds from over 150,000 CS papers recently published on arXiv.org. With this testbed, users are implicitly incentivized to understand and improve the recommender system powering their feed in order to receive more interesting papers. Note further that each user is a task expert, since the users determine their own preferences.

### Neural Recommender

To generate individual recommendations, we utilize a neural model consisting of a linear SVM on top of neural paper embeddings pre-trained on a similar papers task (Cohan et al. 2020). Each paper is represented by the first vector (i.e., the [CLS] token typically chosen for text classification) after encoding the paper title and abstract using SciBERT (Beltagy, Lo, and Cohan 2019). The neural embedding model is finetuned on a triplet loss $L = \max(0, v_i^T v_+ - v_i^T v_- + m)$ where $m$ is a margin hyperparameter and $v_i$, $v_+$ and $v_-$ are the vectors representing a query paper, a similar paper to the query paper, and a dissimilar paper to the query paper, respectively. The similar paper triples are heuristically defined using citations from the SEMANTIC SCHOLAR corpus (Amir et al. 2018), treating cited papers as more similar than un-cited papers. Recommendations are generated by training the model on a user’s annotation history, with additional negative examples randomly drawn from the full corpus of unannotated papers.

A user begins the process of curating their feed by either selecting a specific arXiv CS category or issuing a keyword search and then rating a handful of the resulting papers. A feed consists of a list of recommended papers sorted by predicted recommendation score (see Figure 4). Each paper can be rated using traditional “More like this” or “Less like this” buttons underneath each paper description.

### Implementation of Explanations and Feedback

The UI for Semantic Sanity (Figure 4) displays a list of recommended papers and adorns each with an explanation comprising four terms; to the left of each term are thumbs-up and thumbs-down buttons, enabling the user to act on the explanation and indicate if they would like to see more or fewer papers related to that term. The explanatory terms are generated using a simple, explanatory model (LIMEADE’s EXPLAIN function), which we implement as a linear model over uni- and bigram features. Specifically, we select the 20,000 features with the highest term frequency across our corpus. Our approach of using a post-hoc explanatory model is similar to that used by LIME, except to enable real-time performance our explanatory model is trained as a global, rather than local, approximation of the neural model (Ribeiro, Singh, and Guestrin 2016).[2]

Given the explanatory model, LIMEADE’s DISPLAY function chooses explanations to display by computing each term’s contribution to the output of the linear model for the given paper, which is equal to the product of the term’s TF-IDF value for the paper with the term’s feature weight in the linear model. Next to each explanatory term are thumbs up and down buttons (see Figure 4). When the user provides feedback with these buttons, LIMEADE generates pseudo-examples and retrains the neural recommender. We use a generative approach within GETINSTANCE that leverages the unlabeled pool of papers. We select the top 100 papers from the full corpus with the highest TF-IDF value for the feedback term and generate a single synthetic pseudoeample (i.e., we use $k = 1$) equal to the centroid of these papers’ embeddings. The example is appended to the user’s history and labeled with the user’s annotation of the term (+/-).

### Online Traffic

Since the launch of Semantic Sanity, users with accounts have constructed 2,478 feeds and have logged 21,713 paper annotations and 1,320 topic annotations (we note that annotating topics was only possible after the LIMEADE-based implementation was introduced, five months after the initial launch of Semantic Sanity). Overall, these usage metrics indicate high traffic on the public website. Later in this paper, we analyze a subset of the user logs as part of our evaluation.

### LIMEADE Improves Recommendation Quality

Does using LIMEADE to tune a recommender improve the quality of the system’s rankings in practice? We investigated this question using the log data of Semantic Sanity deployed online. Specifically, we compiled a data set of 1,636 rated papers across 30 feeds, where each feed had at least one annotated explanation (the average number of annotations for these feeds was 4.4 terms). We evaluate two recommenders: a baseline ranker that uses only the rated papers, testing on an early prototype revealed that generating explanations for a feed using LIME was too computationally expensive, since LIME requires sampling nearby instances and training a model for each recommendation on the page; this latency negatively impacted the recommendation experience. Note, in a LIME-style approach to instance-level explanation, one needs to either sample neighbors around the instance (which may be quite distant) or generate synthetic examples, which in our case requires generating paper embeddings on the fly. Both approaches introduced unacceptable latency.
Figure 4: The UI for a feed in Semantic Sanity. Under each paper, the system presents four terms to explain why it was recommended and solicits feedback with “Rate Paper Topics” — by clicking thumbs up or down, the user can request that the feed include more or less of the specified topic.

and a LIMEADE ranker that uses both the rated papers and the annotated terms. We evaluate at three different training sizes (2, 5, and 10 labeled papers), and for each feed and size compute the average normalized discounted cumulative gain (NDCG) ranking performance for up to ten sampled training sets. The average of these statistics across feeds is our final evaluation measure.

Table 2: Simulated evaluation of ranking performance (NDCG) based on log data from actual usage in case study 2. LIMEADE improves performance over a baseline that does not use the annotated explanations.

| Number of labeled papers | Baseline | LIMEADE |
|--------------------------|----------|---------|
| 2                        | 0.884    | 0.899   |
| 5                        | 0.901    | 0.910   |
| 10                       | 0.908    | 0.913   |

Table 2 shows that LIMEADE does improve performance over the baseline, but the benefits of the annotated explanations diminish as the number of rated papers increases. The individual differences shown in the table are not statistically significant, but the aggregate performance over all three sizes shows LIMEADE performing significantly better than the baseline (p-value 0.017, two-tailed paired t-test, after Holm-Bonferoni correction). Notably, LIMEADE with 2 and 5 annotated papers performs comparably to the baseline with 5 and 10 annotated papers, respectively, suggesting that during the early phases of tuning a recommender, term annotations can serve as a low-cost substitute for assessing additional papers. Experiments with more users and feeds are necessary to further validate these claims.

**User Study**

**Experimental Setup**

In order to evaluate the effectiveness of our LIMEADE-based system for recommending papers with real users, we performed an in-person user study. We recruited 21 participants through a public university’s computer science email lists. All participants were adults who reported experience with reading computer science research papers in our screening questionnaire. Each session lasted one hour, and each participant was compensated with a $25 Amazon gift card.

Participants were asked to curate feeds of computer science papers pertaining to a topic of their choice using two different recommendation user interfaces (UIs), one that used LIMEADE to provide tunable explanations, and one that did not present explanations (the baseline); other than this difference, the UIs were the same. The participants were asked to choose a topic that they were interested in following over time as new papers are added to the arXiv, but not so general that it is already covered by an existing arXiv CS category (e.g., artificial intelligence). Once a topic was selected, each participant was asked to name the desired feed, which served as the goal for curation using both UIs.

Each participant began curation by selecting exactly three seed papers that were then used to initialize the feeds in both UIs. Both systems surfaced the same initial recommendations and thus had identical initial states for each participant. Each participant was then presented with one of the two UIs and given instructions on how to use it. 11 participants received the baseline system first, and 10 received the LIMEADE system first. They were then presented with the second UI. For both UIs, the participants were told to use as many or as few annotations as desired until their feed was curated to their liking, or a maximum of 10 minutes.
Table 3: Among 21 participants, most prefer our system over the baseline when prompted with these questions. (*) indicates a statistically significant result under a two-sided paired t-test against a null hypothesis of zero mean difference between the systems.

| Likert scale rating | Baseline | LIMEADE | p-value |
|---------------------|----------|---------|---------|
| Overall system      | 3.38 ± 0.59 | 3.85 ± 0.57* | 0.043   |
| Would use again?    | 3.38 ± 1.16 | 3.90 ± 0.94 | 0.257   |

Table 4: Mean ± Standard Deviation of 21 participant ratings of each system. Ratings were on a scale from 1 (worst/no) to 5 (best/yes). (*) indicates a statistically significant result under a two-sided paired t-test against a null hypothesis of zero mean difference between the systems.

| Likert scale rating | Baseline | LIMEADE | p-value |
|---------------------|----------|---------|---------|
| Overall system      | 3.38 ± 0.59 | 3.85 ± 0.57* | 0.043   |
| Would use again?    | 3.38 ± 1.16 | 3.90 ± 0.94 | 0.257   |

Blind Ratings of Recommendations We also investigated whether the topic-level feedback provided by LIMEADE measurably increased the quality of participants’ feed. We showed participants the top 20 recommendations generated by both systems on the held-out corpus of papers and measured their ratings. Specifically, we computed the discounted cumulative gain (DCG) and average precision (AP), common metrics for assessing recommendation feed quality. For DCG, we observe a mean difference of 0.259 in favor of the baseline system recommendations; however, the corrected p-value for the two-sided, paired t-test for mean differences is 0.218, indicating no significant difference in feed quality between the two systems. For AP, we observe a mean difference of 0.0412 in favor of the baseline system, with a corrected p-value of 0.257, also indicating lack of significant difference in quality under AP. Based on the constructive feedback that we received, we speculate that this result could be improved by making implementation-specific adjustments to Semantic Sanity.

Feed Curation Time In analyzing the times required by each participant to complete feed curation using the two systems, we observe that eight participants finished feed curation with the baseline system first, seven finished with the LIMEADE system first, and the remaining six utilized all ten minutes for the curation of both systems.

Paper and Topic Ratings To explore the breakdown of participants’ rating habits with the baseline system and the LIMEADE system, we present Figure 5. In the top plot in Figure 5, we observe that participants displayed a high degree of variance in the number of ratings applied during feed curation, ranging from 7 annotations to 61 annotations with the baseline system. Comparing the total number of annotations made using the system with LIMEADE vs. the number of annotations made with the baseline, we find a best-fit slope of 0.913. This suggests that the participants made approximately the same number of annotations across both systems.

In the bottom plot, we observe that there is significant diversity in how participants applied topic annotations, ranging from 2 annotations to 27. However, most participants utilized a combination of paper and topic ratings, with more paper ratings than topic ratings on average. Interestingly, five out of the twenty-one participants provided more negative paper ratings than positive ones in the baseline; when presented with the LIMEADE affordances, no participants provided more negative paper ratings than positive ones, but four participants applied more negative topic ratings than positive ones.

Qualitative feedback

We analyze participants’ text responses and provide a sample of quotes that complement the quantitative results. Overall, participants found the tuning affordance granted by our system helpful: “The explanations here were especially useful in their capacity as decisions rather than just explanations.”

For all statistical significance tests, we report adjusted p-values using the Holm-Bonferroni procedure for multiple comparisons (Holm 1979) in R’s p.adjust library (R Core Team 2018).
Figure 5: Scatter plots showing (top) the total number of annotations used to curate a feed with LIMEADE (paper annotations + topic annotations) vs. the number of baseline paper annotations per user, and (bottom) the number of topic annotations vs. the number of paper annotations in the LIMEADE system. Most participants used LIMEADE-powered topic-level feedback as well as paper-level feedback.

In particular, participants stressed the importance of the LIMEADE affordance as a filtering mechanism:

“The topics feature was excellent, because there are many papers which cover *some* topics I like but also some that I don’t, and this let me pick that out.”

The constructive feedback received in the qualitative responses illustrate a number of implementation-specific improvements that could be made to Semantic Sanity. The most common category of constructive feedback concerned the quality of terms in the explanations, mentioned by 10 participants. Though we utilized stemming to eliminate these redundancies in each paper explanation, we did not eliminate synonyms from the list of terms. For example, three of the ten participants specifically requested that abbreviations in explanations be removed or linked to full terms. These issues reflect the negative consequences of utilizing 20,000 TF-IDF terms for our explanatory model featureization. In addition, five of these users also stated that the terms were too general. We speculate that the term quality in the explanations negatively impacted the users’ ability to tune the model via the LIMEADE affordance.

Similarly, three participants directly addressed what we term the explanation-action tradeoff in the next section, noting that the lack of diversity of terms in the explanations was limiting. One participant commented: “After a few minutes, almost all the same terms that I had liked were coming up, so there were few new terms for me to thumbs up or down. I think if the system could focus on bringing up relevant papers that have a new term or two to which I can react, that would make the curation even better.” This suggests tuning the system to favor more explanation diversity even more than we did in our initial implementation.

Interestingly, two users believed that the set of topics surfaced was too restrictive, one thought that the terms were too diverse, and one thought the diversity was a good feature. This provides some evidence that different users have different preferences for explanation diversity, suggesting that it should perhaps be tuned in a user-specific manner. Additionally, four participants commented on topic annotation strength, all of whom indicated that it was too potent. Based on this feedback, we reduced the annotation strength in our application following the evaluation.

**Discussion**

**Exposing the Explanation-Action Tradeoff**

Semantic Sanity chooses explanations to display by computing each term’s contribution to the output of the linear model for the given paper, which is equal to the product of the term’s TF-IDF value for the paper with the term’s feature weight in the linear model. The canonical explanation choice is to surface the terms with the highest-magnitude contributions in the linear model (Ribeiro, Singh, and Guestrin 2016); we call this a greedy approach. However, comments from early users of our paper recommender indicated that there is a tradeoff between using the greedy approach and providing affordances for feedback, which we call the explanation-action tradeoff. In particular, user action on a feature will lead the model to place increasing importance on it and correlated features. With the greedy approach, these terms will begin to dominate the explanations, limiting the number of unique explanation terms and thus opportunities for feedback. For example, ‘thumbs-up’ing the term “fairness” causes papers about fairness to rise in the feed; under the greedy approach, these papers will contain the term “fairness” in their explanations, thereby crowding out new terms for the user to act on. Conversely, a diversity-biased strategy would present more unique terms to provide more opportunities for feedback, but would diverge from using the terms with the highest contributions to the explantory model.

Based on this feedback, our final implementation of DIS-PLAY uses a diversity-biased approach that samples explanatory features controlled by a parameter $\gamma$. We sample terms proportionally to the magnitude of term contribution, raised to the $\gamma$ power (higher values of $\gamma$ result in a more greedy approach; lower values increase diversity). We selected $\gamma = 4$ for our implementation. To further reduce term redundancy in each recommended paper’s explanation, we used
Figure 6: A scatter plot showing the number of unique explanation terms in the first page of the feed vs. the number of actions taken by the user to tune their feed. Orange dots correspond to diversity-biased explanations currently used in the system. Blue dots correspond to greedy explanations, where the most important terms are surfaced without stochasticity. The size of each dot corresponds to the number of feeds in that bin. Note that greedy explanations (blue) display a stronger negative correlation between unique terms and term annotations than diversity-biased explanations (orange). Thus, the greedy approach limits opportunities for tuning with topics as the feed curation process evolves, while the diversity-biased approach continues to facilitate tuning with topics.

the Python NLTK PorterStemmer (Loper and Bird 2002) to deduplicate terms with the same stems (e.g., “fair” and “fairness”) from each explanation.

To illustrate the impact of the explanation-action tradeoff and the distinction between our diversity-biased approach and the canonical greedy method, we perform an analysis on the logs of 300 users’ feeds from Semantic Sanity’s online deployment. For each user, we compute (i) the total number of actions the user has taken on displayed explanatory terms, and (ii) the number of unique explanation terms among the latest top eight recommended papers under our diversity-biased DISPLAY implementation. We then repeat (ii) but with DISPLAY with $\gamma = \infty$ to simulate what explanatory terms the users would see today under a greedy approach.

In accordance with the explanation-action tradeoff, we observe in Figure 6 that the number of unique explanation terms (i.e. tuning affordances) tends to be lower under a greedy approach. Furthermore, this effect grows stronger as users tune their feeds to be increasingly specific to a particular topic. In contrast, the number of affordances remains relatively constant under our diversity-biased approach. Though some explanation terms with lower contribution weight are included within the explanatory model, our diversity-biased approach thus successfully mitigates the crowding effect observed with the greedy approach.

The explanation-action tradeoff is related to, but distinct from, the classical explore-exploit tradeoff faced by recommender systems and other machine learners (Sutton and Barto 2018). The explore-exploit tradeoff entails deliberately passing up a known reward in the hopes of learning more about the reward structure in order to have better long-term gains. Thus, the explore-exploit tradeoff encourages taking a chance in executing an action in the hopes that it will provide a big reward, leading to frequent execution of the action in the future. The explanation-action tradeoff is similar to the explore-exploit tradeoff, in the sense that it entails deliberately declining to provide the most accurate explanation in the hope that providing an affordance for the user to execute a feedback action will lead to better long term recommendations. However, with the explanation-action tradeoff, even if the system is fortunate when taking a chance by providing a less faithful explanation that successfully solicits user feedback, the system will never want to repeat the specific explanation-action in the future.

Decoupling the Effect of Explanations & Tunability

Previous studies have shown that users prefer recommendations with explanations over recommendations alone (Tintarev and Masthoff 2007; Zhang and Chen 2018). In our user study, we did not include an “explanations only” baseline, which would have helped to isolate the contribution of explanations in the preference for our LIMEADE
system among participants. However, we did analyze the user study results post-hoc to investigate this question. In particular, we studied the results in Tables 3 and 4 in order to assess whether participants’ self-reported preferences for our LIMEADE system over the baseline system correlated with utilization of the LIMEADE affordance for rating topics. The participants who voted LIMEADE higher on trust, transparency, intuitiveness, and confidence in not missing papers performed 5.4, 4.6, −0.5, and 3.8 more topic annotations, on average, than those who voted the baseline higher, respectively. This suggests that the positive outcomes for those metrics were not a result of the explanations alone, but were influenced by the tunability of LIMEADE.

In Figure 7, we investigate how the number of topic ratings used by each participant varies as a function of their Likert scale ratings in Table 4. We find that a higher overall rating of our LIMEADE system and a higher self-reported likelihood of using our LIMEADE system in the future are correlated with using more topic annotations. This indicates that more usage of the LIMEADE affordance correlates with a more positive perception of the LIMEADE system.

Conclusion & Future Work
To be effective partners in a human-AI team, an AI system must be able to explain its decisions and allow the human to correct its behavior in terms of that explanation. While intelligible classifiers such as GAMs support explanation-based tuning, and post-hoc methods such as LIME provide explanations for opaque ML models, we present a novel method for updating the opaque model using the approximate features used in the explanation. Furthermore, we are the first to show such a method is effective with user studies. In our first case study, we successfully implemented LIMEADE in the context of image classification. In our second case study, we utilized Semantic Sanity, a publicly-available computer science research paper recommender powered by LIMEADE, to conduct a user study and showed that participants strongly prefer our tunable system over the baseline according to metrics such as trust, perceived control, and overall satisfaction. Examining the logs of 300 users of Semantic Sanity, we have uncovered a fundamental tension between canonical explanation approaches that greedily select the most influential features and those that provide the best affordances for tuning.

From search & recommendation to image recognition to medical diagnosis, opaque machine learners are ubiquitous. End-users deserve new methods for tuning and adjusting these machine learning systems. In this paper, we have presented LIMEADE, a novel framework for tuning an opaque model based on an explanation of the model’s prediction, and have evaluated it with real users. We hope that this work will contribute to opening a new direction of research in human-AI interaction devoted to this challenging and pressing problem. Much work remains to be done. It would be useful to evaluate the tradeoffs between representing a LIMEADE update as a single, potentially off-manifold pseudo-example versus a retrieved sample of previously unlabeled data points from the underlying distribution. Second, in the image domain, it would be valuable to explore how to enhance LIMEADE to accept human feedback at varying levels of granularity. Third, in the text domain, it is important to experiment with aspects of LIMEADE that benefit from personalization, such as tuning potency and explanation diversity. Finally, while LIMEADE works for any opaque learner, there may be better ways to tune specific types of opaque learners (e.g., a specific architecture of deep neural network); we should try to quantify whether domain-specific methods are more effective.

Acknowledgments
The authors would like to thank Sam Skjonsberg and Daniel King for their help in setting up the initial Semantic Sanity prototype; Matt Latzke for his interface design work; Chelsea Haupt and Sebastian Kohlmeier for their management of platform development; Alex Schokking for his work on scaling Semantic Sanity for public launch; Arman Cohan and Sergey Feldman for providing the neural paper embeddings; Yogi Chandrasekhar and Chris Wilhelm for their guidance with implementing LIMEADE for Semantic Sanity; Chris Wilhelm for his help in implementing functionality necessary for the user study; Ani Kembhavi for his advice regarding LIMEADE in the image domain; and Iz Beltagy,
References

Ahn, J.-w.; Brusilovsky, P.; Grady, J.; He, D.; and Syn, S. Y. 2007. Open User Profiles for Adaptive News Systems: Help or Harm? In WWW ’07, 11–20. ACM. ISBN 978-1-59593-654-7. doi:10.1145/1242572.1242575. http://doi.acm.org/10.1145/1242572.1242575

Ahn, J.-w.; Brusilovsky, P.; and Han, S. 2015. Personalized Search: Reconsidering the Value of Open User Models. In IUI ’15, 202–212. ACM. ISBN 978-1-4503-3306-1. doi:10.1145/2678025.2701410. http://doi.acm.org/10.1145/2678025.2701410

Amershi, S.; Fogarty, J.; and Weld, D. 2012. Regroup: Interactive Machine Learning for On-demand Group Creation in Social Networks. In CHI ’12, 21–30. ACM. ISBN 978-1-4503-1015-4. doi:10.1145/2207676.2207680. http://doi.acm.org/10.1145/2207676.2207680

Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Suh, J.; Iqbal, S.; Bennett, P.; Inkpen, K.; and et al. 2019. Guidelines for Human-AI Interaction. In CHI ’19, ACM. ISBN 9781450359702. doi:10.1145/3290605.3300233. https://doi.org/10.1145/3290605.3300233

Ammar, W.; Groeneveld, D.; Bhagavatula, C.; Beltagy, I.; Crawford, M.; Downey, D.; Dunkelberger, J.; and et al. 2018. Construction of the Literature Graph in Semantic Scholar. In NAACL-HLT ’18, 84–91. ACL. doi:10.18653/v1/N18-3011. https://www.aclweb.org/anthology/N18-3011

Bakalov, F.; Meurs, M.-J.; König-Ries, B.; Sateli, B.; Witte, R.; Butler, G.; and Tsang, A. 2013. An Approach to Controlling User Models and Personalization Effects in Recommender Systems. In IUI ’13, 49–56. ACM. ISBN 978-1-4503-1965-2. doi:10.1145/2449396.2449405. http://doi.acm.org/10.1145/2449396.2449405

Bau, D.; Liu, S.; Wang, T.; Zhu, J.-Y.; and Torralba, A. 2020. Rewriting a Deep Generative Model. In Proceedings of the European Conference on Computer Vision (ECCV).

Beel, J.; Gipp, B.; Langer, S.; and Breitinger, C. 2016. Research-paper recommender systems: a literature survey. International Journal on Digital Libraries 17(4): 305–338. ISSN 1432-1300. doi:10.1007/s00799-015-0156-0. https://doi.org/10.1007/s00799-015-0156-0

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In EMNLP ’19.

Bhagavatula, C.; Feldman, S.; Power, R.; and Ammar, W. 2018. Content-Based Citation Recommendation. In ACL ’18, 238–251. New Orleans, LA: ACL. doi:10.18653/v1/N18-1022. https://www.aclweb.org/anthology/N18-1022

Bostandjiev, S.; O’Donovan, J.; and Höllerer, T. 2012. TasteWeights: A Visual Interactive Hybrid Recommender System. In RecSys ’12, 35–42. ACM. ISBN 978-1-4503-1270-7. doi:10.1145/2365952.2365964. http://doi.acm.org/10.1145/2365952.2365964

Bostandjiev, S.; O’Donovan, J.; and Höllerer, T. 2013. LinkedVis: exploring social and semantic career recommendations. In IUI ’13, 107. ACM. ISBN 978-1-4503-1965-2. doi:10.1145/2449396.2449412. http://dl.acm.org/citation.cfm?doid=2449396.2449412

Bruns, S.; Valdez, A. C.; Greven, C.; Ziefle, M.; and Schroeder, U. 2015. What Should I Read Next? A Personalized Visual Publication Recommender System. In Human Interface and the Management of Information. Information and Knowledge in Context, 89–100. Springer. ISBN 978-3-319-20618-9.

Brusilovsky, P.; de Gemmis, M.; Felfernig, A.; Lops, P.; O’Donovan, J.; Semeraro, G.; and Willemsen, M. C. 2020. Interfaces and Human Decision Making for Recommender Systems. In RecSys ’20, 613–618. ISBN 9781450375832. https://doi.org/10.1145/3383333.3411539

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 KDD ’15, 1721–1730. ACM. ISBN 9781450336642. doi:10.1145/2783258.2788613. https://doi.org/10.1145/2783258.2788613

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers.

Cramer, H.; Evers, V.; Ramilal, S.; van Someren, M.; Rutledge, L.; Stash, N.; Aroyo, L.; and Wielinga, B. 2008. LinkedVis: exploring social and semantic career recommendations. In IUI ’08, 172–181. ACM. ISBN 9781605581644. URL http://doi.acm.org/10.1145/1390334.1390436

Doshi-Velez, F.; and Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. URL http://arxiv.org/abs/1702.08608

Druck, G.; Mann, G.; and McCallum, A. 2008. Learning from Labeled Features Using Generalized Expectation Criteria. In SIGIR ’08, 595–602. ISBN 9781605581644. URL https://dl.acm.org/doi/10.1145/1390334.1390436
Ekstrand, M.; Kannan, P.; Stemper, J.; Butler, J.; Konstan, J.; and Riedl, J. 2010. Automatically Building Research Reading Lists. In RecSys ’10, 159–166. ACM. ISBN 9781605589060. doi:10.1145/1864708.1864740. URL https://doi.org/10.1145/1864708.1864740

Godbole, S.; Harpale, A.; Sarawagi, S.; and Chakrabarti, S. 2004. Document Classification Through Interactive Supervision of Document and Term Labels. In Boulicaut, J.-F.; Esposito, F.; Giannotti, F.; and Pedreschi, D., eds., PKDD ‘04, 185–196. ISBN 978-3-540-30116-5.

Gretarsson, B.; O’Donovan, J.; Bostandjiev, S.; Hall, C.; and Höllerer, T. 2010. SmallWorlds: Visualizing Social Recommendations. Computer Graphics Forum 29(3): 833–842. doi:10.1111/j.1467-8659.2009.01679.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01679.x

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A Survey of Methods for Explaining Black Box Models. ACM Comput. Surv. 51(5). ISSN 0360-0300. doi:10.1145/3236009. URL https://doi.org/10.1145/3236009

Harper, F. M.; Xu, F.; Kaur, H.; Condiff, K.; Chang, S.; and Terveen, L. 2015. Putting Users in Control of their Recommendations. In RecSys ’15, 3–10. ACM. ISBN 978-1-4503-3692-5. doi:10.1145/2792838.2800179. URL http://dl.acm.org/citation.cfm?doid=2792838.2800179

He, C.; Parra, D.; and Verbert, K. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. Expert Systems with Applications 56: 9–27. ISSN 09574174. doi:10.1016/j.eswa.2016.02.013. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417416300367

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385.

He, Q.; Pei, J.; Kifer, D.; Mitra, P.; and Giles, L. 2010. Context-aware Citation Recommendation. In WWW ’10, 421–430. ACM. ISBN 978-1-60558-799-8. doi:10.1145/1772690.1772734. URL http://doi.acm.org/citation.cfm?id=1772690.1772734

Herlocker, J.; Konstan, J.; and Riedl, J. 2000. Explaining collaborative filtering recommendations. In CSCW ’00, 241–250. ACM. ISBN 978-1-58113-222-9. doi:10.1145/358916.358995. URL http://portal.acm.org/citation.cfm?doid=358916.358995

Holm, S. 1979. A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics 6(2): 65–70. ISSN 03036898. 14679469.

Jin, Y.; Tintarev, N.; and Verbert, K. 2018. Effects of personal characteristics on music recommender systems with different levels of controllability. In RecSys ’18, 13–21. ACM. ISBN 978-1-4503-5901-6. doi:10.1145/3240323.3240358. URL http://dl.acm.org/citation.cfm?doid=3240323.3240358

Karpathy, A. 2015. Arxiv Sanity Preserver. URL http://www.arxiv-sanity.com/

Knijnenburg, B.; Reijmers, N.; and Willemsen, M. 2011. Each to His Own: How Different Users Call for Different Interaction Methods in Recommender Systems. In RecSys ’11, 141–148. ACM. ISBN 978-1-4503-0683-6. doi:10.1145/2043932.2043960. URL http://doi.acm.org/10.1145/2043932.2043960

Kulesza, T.; Burnett, M.; Wong, W.-K.; and Stumpf, S. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In IUI ’15, 126–137. ACM. ISBN 978-1-4503-3306-1. doi:10.1145/2678025.2701399. URL http://doi.acm.org/10.1145/2678025.2701399

Kulesza, T.; Stumpf, S.; Burnett, M.; and Kwan, I. 2012. Tell me more?: the effects of mental model soundness on personalizing an intelligent agent. In IUI ’12, 1. ACM. ISBN 978-1-4503-1015-4. doi:10.1145/2207676.2207678. URL http://dl.acm.org/citation.cfm?doid=2207676.2207678

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 VL/HCC ’13, 3–10. ISSN 1943-6092. doi:10.1109/VLHCC.2013.6645235.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2014. Microsoft COCO: Common Objects in Context. URL http://arxiv.org/abs/1405.0312

Liu, B.; Li, X.; Lee, W. S.; and Yu, P. S. 2004. Text Classification by Labeling Words. In AAAI ’04, 425–430.

Liu, F.; and Avci, B. 2019. Incorporating Priors with Feature Attribution on Text Classification. In ACL ’19, 6274–6283. ACL.

Loepp, B.; Barbu, C.-M.; and Ziegler, J. 2016. Interactive Recommending: Framework, State of Research and Future Challenges. In Proceedings of the 1st Workshop on Engineering Computer-Human Interaction in Recommender Systems, 3–13. URL http://ceur-ws.org/V ol-1705/02-paper.pdf

Loper, E.; and Bird, S. 2002. NLTK: The Natural Language Processing Toolkit. In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. ACL ’02.

Lou, Y.; Caruana, R.; and Gehrke, J. 2012. Intelligible Models for Classification and Regression. In KDD ’12, 150–158. ACM. ISBN 9781450314626. doi:10.1145/
Advances in Neural Information Processing Systems, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., Dialog Explanations of Neurons. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., Advances in Neural Information Processing Systems, volume 33, 17153–17163. Curran Associates, Inc. URL http://proceedings.neurips.cc/paper/2020/file/c74956f0b3bba48ed6ce977af60727275-Paper.pdf.

Lundberg, S.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In NIPS ’17, 4765–4774. Curran Associates, Inc. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Mu, J.; and Andreas, J. 2020. Compositional Explanations of Neurons. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., Advances in Neural Information Processing Systems, volume 33, 17153–17163. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2020/file/c74956f0b3bba48ed6ce977af60727275-Paper.pdf.

O’Donovan, J.; Smyth, B.; Gretarsson, B.; Bostandjiev, S.; and Höllerer, T. 2008. PeerChooser: Visual Interactive Recommendation. In CHI ’08, 1085–1088. ACM. ISBN 978-1-60558-011-1. doi:10.1145/1357054.1357222. URL http://doi.acm.org/10.1145/1357054.1357222.

Parra, D.; and Brusilovsky, P. 2015. User-controllable personalization: A case study with SetFusion. International Journal of Human-Computer Studies 78: 43–67. ISSN 10715819. doi:10.1016/j.ijhcs.2015.01.007. URL https://linkinghub.elsevier.com/retrieve/pii/S1071581915000208.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12: 2825–2830.

Pu, P.; and Chen, L. 2006. Trust Building with Explanation Interfaces. In IUI ’06, 93–100. ACM. ISBN 1-59593-287-9. doi:10.1145/1111449.1111475. URL http://doi.acm.org/10.1145/1111449.1111475.

Pu, P.; Chen, L.; and Hu, R. 2011. A User-centric Evaluation Framework for Recommender Systems. In RecSys ’11, 157–164. ACM. ISBN 978-1-4503-0530-6. doi:10.1145/2043932.2043962. URL http://doi.acm.org/10.1145/2043932.2043962.

R Core Team. 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. URL https://www.R-project.org/.

Raghavan, H.; and Allan, J. 2007. An Interactive Algorithm for Asking and Incorporating Feature Feedback into Support Vector Machines. In SIGIR ’07, 79–86. ISBN 9781595939577. URL https://dl.acm.org/doi/10.1145/1277741.1277758.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. “Should I Trust You?” Explaining the Predictions of Any Classifier. In KDD ’16, 1135–1144. ACM. ISBN 978-1-4503-4232-2. doi:10.1145/2939672.2939778. URL http://doi.acm.org/10.1145/2939672.2939778.

Rieger, L.; Singh, C.; Murdoch, W.; and Yu, B. 2020. Interpretations are Useful: Penalizing Explanations to Align Neural Networks with Prior Knowledge. In III, H. D.; and Singh, A., eds., Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, 8116–8126. PMLR. URL http://proceedings.mlr.press/v119/rieger20a.html.

Rosenthal, S.; and Dey, A. 2010. Towards Maximizing the Accuracy of Human-Labeled Sensor Data. In IUI ’10, 259–268. ISBN 9781605585154.

Ross, A. S.; Hughes, M. C.; and Doshi-Velez, F. 2017. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In IJCAI ’17, 2662–2670. doi:10.24963/ijcai.2017/371. URL https://doi.org/10.24963/ijcai.2017/371.

Schaffer, J.; Höllerer, T.; and O’Donovan, J. 2015. Hypothetical Recommendation: A Study of Interactive Profile Manipulation Behavior for Recommender Systems. URL https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS15/paper/view/10444.

Sehramowski, P.; Stammer, W.; Teso, S.; Brugger, A.; Shao, X.; Luigs, H.-G.; Mahlein, A.-K.; and Kersting, K. 2020. Making deep neural networks right for the right scientific reasons by interacting with their explanations.

Simard, P.; Amershi, S.; Chickering, D.; Pelton, A.; Ghoshari, S.; Meek, C.; Ramos, G.; Suh, J.; Vervey, J.; Wang, M.; and Wernsing, J. 2017. Machine Teaching: A New Paradigm for Building Machine Learning Systems. URL http://arxiv.org/abs/1707.06742.

Sinha, A.; Shen, Z.; Song, Y.; Ma, H.; Eide, D.; Hsu, B.-J. P.; and Wang, K. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In WWW ’15, 243–246. ACM. ISBN 978-1-4503-3473-0. doi:10.1145/2740908.2742839. URL http://doi.acm.org/10.1145/2740908.2742839.

Sutton, R. S.; and Barto, A. G. 2018. Reinforcement learning: an introduction. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, second edition edition. ISBN 978-0-262-03924-6.

Tintarev, N.; and Masthoff, J. 2007. A Survey of Explanations in Recommender Systems. In 2007 IEEE 23rd International Conference on Data Engineering Workshop, 801–810.

Tintarev, N.; and Masthoff, J. 2011. Designing and Evaluating Explanations for Recommender Systems. In Ricci, F.; Rokach, L.; Shapira, B.; and Kantor, P., eds., Recommender Systems Handbook, 479–510. Boston, MA: Springer. ISBN 978-0-387-85820-3. doi:10.1007/978-0-387-85820-3_15. URL https://doi.org/10.1007/978-0-387-85820-3_15.

Tsai, C.-H.; and Brusilovsky, P. 2018. Beyond the Ranked List: User-Driven Exploration and Diversification of Social Recommendation. In IUI ’18, 239–250. ACM. ISBN 978-1-4503-4945-1. doi:10.1145/3172944.3172959. URL http://doi.acm.org/10.1145/3172944.3172959.
Tsai, C.-H.; and Brusilovsky, P. 2019. Explaining Recommendations in an Interactive Hybrid Social Recommender. In *IUI '19*, 391–396. ACM. ISBN 978-1-4503-6272-6. doi:10.1145/3301275.3302318. URL http://doi.acm.org/10.1145/3301275.3302318.

Tsai, C.-H.; and Brusilovsky, P. 2020. The effects of controllability and explainability in a social recommender system. *User Modeling and User-adapted Interaction* 1–37.

Verbert, K.; Parra, D.; Brusilovsky, P.; and Duval, E. 2013. Visualizing Recommendations to Support Exploration, Transparency and Controllability. In *IUI '13*, 351–362. ACM. ISBN 978-1-4503-1965-2. doi:10.1145/2449396.2449442. URL http://doi.acm.org/10.1145/2449396.2449442.

Vig, J.; Sen, S.; and Riedl, J. 2012. The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM TIIS* 2(3): 1–44. ISSN 21606455. doi:10.1145/2362394.2362395. URL http://dl.acm.org/citation.cfm?doid=2362394.2362395.

Wærn, A. 2004. User Involvement in Automatic Filtering: An Experimental Study. *UMUAI* 14(2): 201–237. ISSN 1573-1391. doi:10.1023/B:USER.0000028984.13876.9b.

Weld, D.; and Bansal, G. 2019. The Challenge of Crafting Intelligible Intelligence. *CACM* 62(6): 70–79. ISSN 0001-0782. doi:10.1145/3282486. URL http://doi.acm.org/10.1145/3282486.

Wu, X.; and Srihari, R. 2004. Incorporating Prior Knowledge with Weighted Margin Support Vector Machines. In *KDD '04*, 326–333. ISBN 1581138881. doi:10.1145/1014052.1014089. URL https://doi.org/10.1145/1014052.1014089.

Zhang, Y.; and Chen, X. 2018. Explainable Recommendation: A Survey and New Perspectives. *Arxiv* URL http://arxiv.org/abs/1804.11192.