Explanation-Based Human Debugging of NLP Models: A Survey

Piyawat Lertvittayakumjorn and **Francesca Toni**
Department of Computing
Imperial College London, UK
{pl1515, ft}@imperial.ac.uk

Abstract

To fix a bug in a program, we need to locate where the bug is, understand why it causes the problem, and patch the code accordingly. This process becomes harder when the program is a trained machine learning model and even harder for opaque deep learning models. In this survey, we review papers that exploit explanations to enable humans to debug NLP models. We call this problem *explanation-based human debugging (EBHD)*. In particular, we categorize and discuss existing works along three main dimensions of EBHD (the bug context, the workflow, and the experimental setting), compile findings on how EBHD components affect human debuggers, and highlight open problems that could be future research directions.

1 Introduction

Explainable AI focuses on generating explanations for AI models as well as for their predictions. It is gaining more and more attention these days since explanations are necessary in several applications, especially in high-stake domains such as healthcare, law, transportation, and finance (Adadi and Berrada, 2018). Some researchers have explored various merits of explanations to humans, such as supporting human decision makings (Lai and Tan, 2019), increasing human trust in AI (Jacovi et al., 2020) and even teaching humans to perform challenging tasks (Lai et al., 2020). On the other hand, explanations can benefit the AI systems as well, e.g., when explanations are used to promote system acceptance (Cramer et al., 2008), to verify the model reasoning (Caruana et al., 2015), and to find potential causes of errors (Han et al., 2020).

In this paper, we review progress to date specifically on how explanations have been used in the literature to enable humans to fix bugs in Natural Language Processing (NLP) models. We refer to this research area as *explanation-based human debugging (EBHD)*, as a general umbrella term encompassing *explanatory debugging* (Kulesza et al., 2010) and *human-in-the-loop debugging* (Lertvittayakumjorn et al., 2020). EBHD is helpful when the training data at hand leads to suboptimal models (due to, for instance, biases and artifacts in the data), and hence human knowledge is needed to verify and improve the trained models. In fact, EBHD is related to three challenging and intertwined issues in NLP: explainability (Danilevsky et al., 2020), interactive and human-in-the-loop learning (Amershi et al., 2014; Wang et al., 2021), and knowledge integration (Kim et al., 2021). Although there are overviews for each of these issues (as cited above), our paper is the first to draw connections among the three towards the final application of model debugging in NLP.

Whereas most people agree on the meaning of the term *bug* in software engineering, various meanings have been ascribed to this term in machine learning (ML) research. For example, Selsam et al. (2017) considered bugs as implementation errors, similar to software bugs, while Cadamuro et al. (2016) defined a bug as a particularly damaging or inexplicable test error. In this paper, we follow the definition of (model) bugs from Adebayo et al. (2020) as contamination in the learning and/or prediction pipeline that makes the model produce incorrect predictions or learn error-causing associations. Examples of bugs include spurious correlation, labelling errors, and undesirable behaviour in out-of-distribution (OOD) testing.

The term *debugging* is also interpreted differently by different researchers. Some consider debugging as a process of identifying or uncovering causes of model errors (Parikh and Zitnick, 2011; Graliński et al., 2019), while others stress that debugging must not only reveal the causes of problems but also fix or mitigate them (Kulesza et al., 2015; Yousefzadeh and O'Leary, 2019). In this

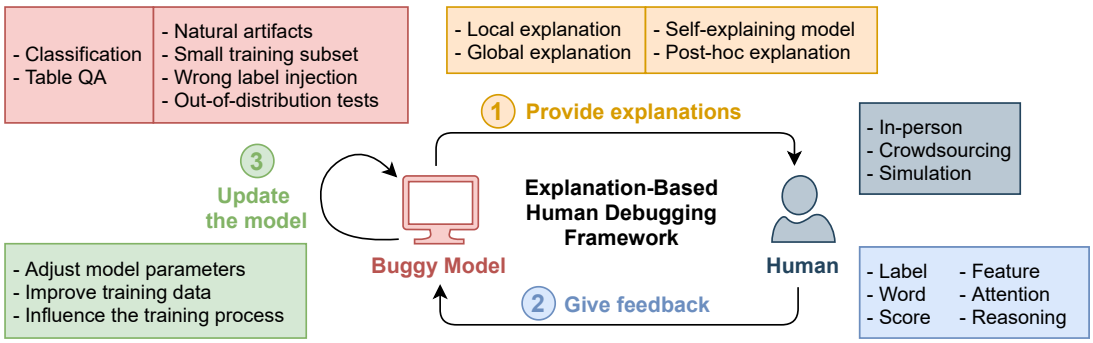

Figure 1: A general framework for explanation-based human debugging (EBHD) of NLP models, consisting of the (potentially) buggy model, the human debugger(s), and a three-step workflow. Colored boxes list examples (considered in the literature) of the options for the corresponding components or steps in the framework.

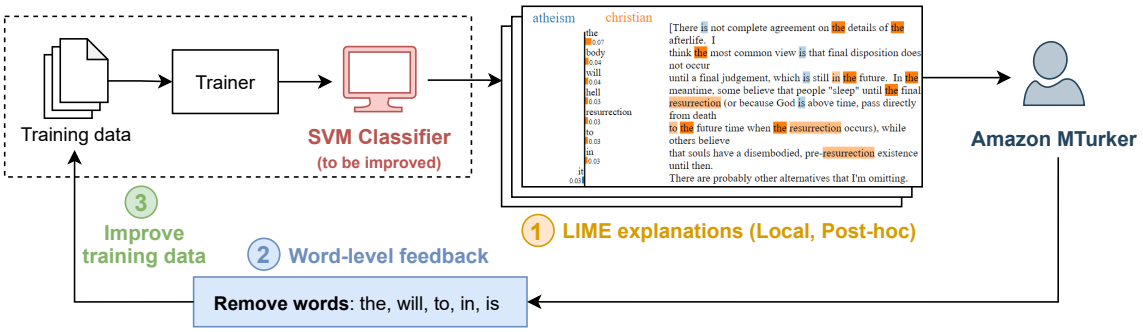

Figure 2: The proposal by Ribeiro et al. (2016) as an instance of the general EBHD framework.

paper, we adopt the latter interpretation.

Overall, to fully set the scope of this paper, we focus on works using explanations of NLP models to reveal the causes of bugs to humans and collect their feedback to update the models/fix the bugs. However, note that, where useful and relevant, we mention related works of the following categories although, strictly speaking, they are not in the main scope of the paper: debugging without explanations (Lourenço et al., 2019; Kang et al., 2018), debugging outside the NLP domain (Ghai et al., 2021; Popordanoska et al., 2020), and works centered on revealing but not fixing bugs (Ribeiro et al., 2020; Krause et al., 2016; Krishnan and Wu, 2017).

Generally, EBHD can be divided into three main steps as shown in Figure 1. Firstly, the model provides interpretable insights about the bugs to human debuggers via explanations. Then, the humans inspect the explanations and give feedback in response. Finally, the feedback is used to update and improve the model. Note that these three steps can be carried out once, as a one-off improvement, or iteratively, depending on how the debugging framework is designed.

As a concrete example, Figure 2 illustrates how Ribeiro et al. (2016) improved an SVM text classifier trained on the 20Newsgroups dataset (Lang, 1995)[1]. This dataset has many artifacts which could make the model rely on wrong words when making predictions, reducing its generalizability. To perform EBHD, Ribeiro et al. (2016) recruited humans from a crowdsourcing platform (i.e., Amazon Mechanical Turk) and asked them to inspect LIME explanations (word attribution scores) for model predictions of ten examples. After that, the humans gave feedback by identifying words in the explanations that should have not got high attribution scores (supposed to be the artifacts). These words were then removed from the training data, and the model was retrained. The process was repeated for three iterations, and the results show that the model generalizes better after every iteration. Using the general EBHD framework in Figure 1, we can break the framework of Ribeiro et al. (2016) into components as depicted in Figure 2. Throughout the paper, when reviewing existing EBHD approaches, we will use the general framework in Figure 1 for analysis, comparison, and discussion.

The rest of this paper is organized as follows. In

[1] http://qwone.com/~jason/20Newsgroups/

Section 2, we explain the choices made by existing works to achieve EBHD of NLP models. This illustrates the current state of the field together with the strengths and limitations of existing works. Naturally, though, to create a successful debugging framework, we cannot neglect the "imperfect" nature of human debuggers, who may not always be an ideal oracle. Therefore, Section 3 compiles relevant human factors which could affect the effectiveness of the debugging process as well as the satisfaction of the human debuggers. After that, we identify open challenges of EBHD for NLP in Section 4 before concluding the paper in Section 5.

2 Categorization of Existing Works

Table 1 summarizes existing works on EBHD of NLP models along three dimensions, including the debugging context (i.e., tasks, models, and bug sources), the workflow (i.e., the three steps in our general framework), and the experimental setting (i.e., the form of human engagement).

2.1 Context

To demonstrate the debugging process, existing works need to set up the bug situation they aim to fix, amounting to the target NLP task, the machine learning model used to carry out that task, and the source of the bug to be addressed.

2.1.1 Tasks

The majority of works focus on text classification (TC) for a variety of specific problems such as e-mail categorization (Stumpf et al., 2009), topic classification (Kulesza et al., 2015; Teso and Kersting, 2019), spam classification (Koh and Liang, 2017), and auto-coding of transcripts (Kulesza et al., 2010). Interestingly, only Cho et al. (2019) did not target TC, but focused instead on table question answering (TQA).

Ghai et al. (2021) suggested that most researchers work on TC because, for this task, it is much easier for human participants (usually lay people) to understand explanations and to give feedback (e.g., which keywords should be added or removed from the list of top features). By contrast, some other NLP tasks require human debuggers to have linguistic knowledge such as part-of-speech tagging, parsing, and machine translation. The need for linguists or experts renders experiments for these tasks more difficult and costly. However, we suggest that there are several tasks which are prone to cause model bugs, not difficult to experiment on (even with lay people), but underexplored in the EBHD setting, including reading comprehension (Jia and Liang, 2017), question answering (Ribeiro et al., 2019), and natural language inference (McCoy et al., 2019; Gururangan et al., 2018).

2.1.2 Models

Early works used Naive Bayes models with bag-of-words (NB) as text classifiers (Kulesza et al., 2009; Stumpf et al., 2009; Kulesza et al., 2010, 2015), which are relatively easy to generate explanations for and to incorporate human feedback into (to be discussed in Section 2.2). Other traditional ML models used include logistic regression (LR) (Koh and Liang, 2017; Teso and Kersting, 2019) and support vector machines (SVM) (Ribeiro et al., 2016), both with bag-of-words features. Recently, Lertvittayakumjorn et al. (2020) focused on convolutional neural networks (CNN) (Kim, 2014) and touched upon bidirectional LSTM networks (Hochreiter and Schmidhuber, 1997). Cho et al. (2019) worked on Neural Operator (NeOp), a multi-layer sequential network with attention supervision (Cho et al., 2018) to solve TQA. While it cannot be denied that the NLP community nowadays is driven by pre-trained language models (Qiu et al., 2020) with several papers studying their behaviors (Rogers et al., 2021; Hoover et al., 2020), there has been only one paper by Yao et al. (2021) using pre-trained language models, i.e., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), as test beds for EBHD.

2.1.3 Bug Sources

Most of the papers in Table 1 experimented on training datasets with natural artifacts (AR), which cause spurious correlation bugs (i.e., the input texts having signals which are correlated to but not the reasons for specific predicted classes) and undermine models' generalizability. Five out of the eleven papers we survey used the 20Newsgroups dataset (Lang, 1995) as a case study, since it has lots of natural artifacts. Apart from classification performance drops, natural artifacts can also cause model biases, as shown in (De-Arteaga et al., 2019; Park et al., 2018) and debugged in (Lertvittayakumjorn et al., 2020; Yao et al., 2021).

In the absence of strong natural artifacts, bugs can still be simulated using several techniques. First, using only a small subset of labeled data (SS) for training could cause the model to exploit

Paper	Context			Workflow				Setting
	Task	Model	Bug sources	Exp. scope	Exp. method	Feed-back	Up-date	
Kulesza et al. (2009)	TC	NB	AR	G,L	SE	LB,WE	M,D	IP
Stumpf et al. (2009)	TC	NB	SS	L	SE	WO	T	IP
Kulesza et al. (2010)	TC	NB	AR	G,L	SE	WO	M,D	IP
Kulesza et al. (2015)	TC	NB	AR	G,L	SE	WO,WE	M	IP
Ribeiro et al. (2016)	TC	SVM	AR	L	PH	WO	D	CS
Koh and Liang (2017)	TC	LR	WL	L	PH	LB	D	SM
Teso and Kersting (2019)	TC	LR	AR	L	PH	WO	D	SM
Cho et al. (2019)	TQA	NeOp	AR	L	SE	AT	T	NR
Lertvittayakumjorn et al. (2020)	TC	CNN	AR,SS,OD	G	PH	FE	T	CS
Smith-Renner et al. (2020)	TC	NB	AR,SS	L	SE	LB,WO	M,D	CS
Yao et al. (2021)	TC	BERT*	AR,OD	L	PH	RE	D,T	NR

Table 1: Overview of existing works on EBHD of NLP models. We use abbreviations as follows: **Task**: TC = Text Classification, TQA = Table Question Answering / **Model**: NB = Naive Bayes, SVM = Support Vector Machines, LR = Logistic Regression, NeOp = Neural Operator, CNN = Convolutional Neural Networks, BERT* = BERT and RoBERTa / **Bug sources**: AR = Natural artifacts, SS = Small training subset, WL = Wrong label injection, OD = Out-of-distribution tests / **Exp. scope**: G = Global explanations, L = Local explanations / **Exp. method**: SE = Self-explaining, PH = Post-hoc method / **Feedback**: LB = Label, WO = Word, WE = Score, FE = Feature, AT = Attention, RE = Reasoning / **Update**: M = Adjust the model parameters, D = Improve the training data, T = Influence the training process / **Setting**: IP = In-person, CS = Crowdsourcing, SM = Simulation, NR = Not reported.

spurious correlation leading to poor performance (Kulesza et al., 2010). Second, injecting wrong labels (WL) to the training data can obviously blunt the model quality (Koh and Liang, 2017). Finally, using out-of-distribution tests (OD) can reveal that the model does not work effectively in the domains that it has not been trained on (Lertvittayakumjorn et al., 2020; Yao et al., 2021). All of these undesirable behaviors require debugging.

2.2 Workflow

This section describes existing works around the three steps of the EBHD workflow in Figure 1, i.e., how to generate and present the explanations, how to collect human feedback, and how to update the model using the feedback. Researchers need to make decisions on these key points harmoniously to create an effective debugging workflow.

2.2.1 Providing Explanations

The main role of explanations here is to reveal misbehavior or irrationality of the model, which sometimes cannot be noticed by looking at the model outputs or the evaluation metrics.

Explanation types. Basically, there are two types of explanations that could be provided to humans. Local explanations (L) explain the predictions by the model for individual inputs. In contrast, global explanations (G) explain the model overall, independently of any specific inputs. It can be seen from Table 1 that most existing works provide local explanations. One reason for this may be that, for complex models, global explanations can hardly reveal details of the model's inner workings in a comprehensible way to users. So, some bugs are imperceptible in such high-level global explanations and then not corrected by the users. For example, the debugging framework *FIND*, proposed by Lertvittayakumjorn et al. (2020), uses only global explanations, and it was shown to work more effectively on significant bugs (such as gender bias in abusive language detection) than on less-obvious bugs (such as dataset shift between product types of sentiment analysis on product reviews).

On the other hand, using local explanations has limitations in that it is hardly possible for users to inspect the explanation of every single example in the training/validation set. Therefore, the use of local explanations requires efficient ways to rank or select examples to explain. For instance, Ribeiro et al. (2016) picked sets of non-redundant local explanations to illustrate the global picture of the model. Instead, Teso and Kersting (2019) leveraged heuristics from active learning to choose unlabeled examples that maximize some informativeness criteria.

Generating explanations. Some ML models, e.g., Naive Bayes, logistic regression, and decision trees, are self-explaining (SE) (Danilevsky et al.,

2020), also referred to as transparent (Adadi and Berrada, 2018), inherently interpretable (Rudin, 2019), or directly interpretable (Arya et al., 2019). Local explanations of self-explaining models can be obtained at the same time as predictions, usually from the process of making those predictions, while the models themselves can serve directly as global explanations. For example, for a Naive Bayes classifier where $P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$, if the task is binary classification with classes y and $\hat{y}$, we could show some x_i $(1 \leq i \leq n)$ with the highest $P(x_i|y) - P(x_i|\hat{y})$ as the local explanation as to why the prediction of $(x_1, \ldots, x_n)$ is the class y. Also, we could consider $P(y)$, $P(\hat{y})$, $P(x_j|y)$ and $P(x_j|\hat{y})$ for any possible feature x_j to be the global explanation. All the papers in Table 1 whose model is Naive Bayes utilized these ideas to generate explanations. Meanwhile, Cho et al. (2019) used attention scores on inputs as explanations. We consider this as a self-explaining method too because the attention scores were obtained from the prediction process.

Besides the self-explaining approach, post-hoc explanation methods (PH) perform additional steps to extract explanations after the model is trained (for a global explanation) or after the prediction is made (for a local explanation). For instance, Ribeiro et al. (2016) and Teso and Kersting (2019) used LIME (Ribeiro et al., 2016) to generate local explanations, showing top-k most important words together with their importance scores.

In addition to LIME, there are other feature importance methods used to generate explanations for EBHD. Yao et al. (2021) used hierarchical explanations (Jin et al., 2020) to visualize the importance of not only words but also phrases in the input. Lertvittayakumjorn et al. (2020) collated LRP explanations (Arras et al., 2016) of several training examples to form a global explanation, in the form of word clouds displaying patterns detected by individual neurons.

In most NLP tasks, especially for text classification, feature importance methods, locating important parts of the input, are the most popular approach for explaining model predictions (Bhatt et al., 2020) as well as for EBHD. There are fewer works studying other forms of local explanations for human debugging such as rule-based (Ribeiro et al., 2018), example-based (Guo et al., 2020), and adversarial-based (Iyyer et al., 2018) explanations, let alone global explanations.

Presenting explanations. The presentation of explanations does not matter much if the human debugger is an ML engineer who understands how to read the explanations. However, if the person providing feedback is an end user lacking technical knowledge, it is important to carefully design the presentation, taking into consideration their background and desires. In the debugging application by Kulesza et al. (2009), lay users were asked to provide feedback to email categorizations predicted by the system. The users were allowed to ask several Why questions (inspired by Myers et al. (2006)) through either the menu bar, or by right-clicking on the object of interest (such as a particular word). Examples include "Why will this message be filed to folder A?", "Why does word x matter to the folder B?". The system then responded by textual explanations (generated using templates), together with visual explanations such as bar plots for some types of questions. All of these made the interface become more user-friendly. In 2015, Kulesza et al. proposed, as desirable principles, that the presented explanations should be sound (i.e., truthful in describing the underlying model), complete (i.e., not omitting important information about the model), but not overwhelming (i.e., remaining comprehensible). However, these principles are challenging especially when working on non-interpretable complex models.

2.2.2 Collecting Feedback

After seeing explanations, humans generally desire to improve the model by giving feedback (Smith-Renner et al., 2020). Some existing works ask humans to confirm or correct machine-computed explanations. Hence, the form of feedback fairly depends on the form of the explanations, and in turn this shapes how to update the model too (discussed in section 2.2.3). For text classification, most EBHD papers asked humans to decide which words (WO) in the explanation (considered important by the model) are in fact relevant or irrelevant (Kulesza et al., 2010; Ribeiro et al., 2016; Teso and Kersting, 2019). Some papers even allowed humans to adjust the word importance scores (WE) (Kulesza et al., 2009, 2015). Meanwhile, feedback at the level of learned features (FE), rather than individual words, was asked in (Lertvittayakumjorn et al., 2020). Additionally, humans may be asked to check the predicted labels (Kulesza et al., 2009; Smith-Renner et al., 2020) or even the ground truth labels (collectively noted as LB in Table 1) (Koh

and Liang, 2017). Targeting the table question answering, Cho et al. (2019) asked humans to identify where in the table and the question the model should focus (AT). This is analogous to identifying relevant words to attend for text classification.

It is likely that identifying important or relevant parts in the input is sufficient to make the model accomplish simple text classification tasks. However, this might not be enough for complex tasks which require reasoning. Recently, Yao et al. (2021) asked humans to provide, as feedback, compositional explanations where linguistic properties and interactions among input words can be specified to show how the humans would reason (RE) about the models' failure cases. To acquire richer information as feedback, their framework requires more expertise from the human debuggers. In the future, it would be interesting to explore how we can collect and utilize other forms of feedback, e.g., natural language feedback (Camburu et al., 2018), ratings of the provided explanations, new training examples (Fiebrink et al., 2009), and other forms of decision rules used by humans (Carstens and Toni, 2017).

2.2.3 Updating the Model

Techniques to incorporate human feedback into the model can be categorized broadly into three approaches.

(1) Directly adjust the model parameters (M). When the model is transparent and the explanation displays the model parameters in an intelligible way, humans can directly adjust the parameters based on their judgements. This idea was adopted by Kulesza et al. (2009, 2015) where humans can adjust a bar chart showing word importance scores, corresponding to the parameters of the underlying Naive Bayes model. Also, human feedback can be used to modify the model parameters indirectly. For example, Smith-Renner et al. (2020) increased a word weight in the Naive Bayes model by 20% for the class that the word supported, according to human feedback, and reduced the weight by 20% for the opposite class (binary classification).

This approach is fast because it does not require model retraining. However, it is important to ensure that the adjustments made by humans generalize well to all examples. Therefore, the system should update the overall results (e.g., performance metrics, predictions, and explanations) in real time after applying any adjustment, so the humans can investigate the effects and further adjust the model parameters (or undo the adjustments) if necessary. This agrees with the correctability principles proposed by Kulesza et al. (2015) that the system should be actionable and reversible, honor user feedback, and show incremental changes.

(2) Improve the training data (D). We can use human feedback to improve the training data and retrain the model to fix bugs. This approach includes correcting mislabeled training examples (Koh and Liang, 2017), assigning noisy labels to unlabeled examples (Yao et al., 2021), removing irrelevant words from input texts (Ribeiro et al., 2016), and creating augmented training examples to reduce the effects of irrelevant words (Teso and Kersting, 2019). As this approach modifies the training data only, it is applicable to any model regardless of the model complexity.

(3) Influence the training process (T). Another approach is to influence the (re-)training process in a way that the resulting model will behave as the human feedback suggests. This approach could be either model-specific (such as attention supervision) or model-agnostic (such as user co-training). Cho et al. (2019) used human feedback to supervise attention weights of the model. Similarly, Yao et al. (2021) added a loss term to regularize explanations guided by human feedback. Stumpf et al. (2009) proposed two techniques. One is constraint optimization, translating human feedback into constraints governing the training process. The other is user co-training, using human feedback as another classifier working together with the main ML model in a semi-supervised learning setting. Lertvittayakumjorn et al. (2020) disabled some learned features deemed irrelevant, according to human feedback, and then re-trained the model, forcing it to use only the remaining features.

2.2.4 Iteration

The debugging workflow – explain, feedback, and update – can be done iteratively to gradually improve the model for most papers in Table 1 where the presented explanation changes after the model update. This is beneficial because humans can fix vital bugs first and finer bugs in later iterations, as we can see from the performance plots in (Ribeiro et al., 2016; Koh and Liang, 2017). However, the interactive process could be susceptible to the phenomenon called *local decision pitfalls* where local improvements for individual predictions could add up to inferior overall performance (Wu et al., 2019).

So, we need to ensure that the update in the current iteration is generally favorable and does not overwrite the good effects of previous updates.

2.3 Experimental Setting

Early works in this area conducted experiments with humans in-person (IP) (Kulesza et al., 2009; Stumpf et al., 2009; Kulesza et al., 2010, 2015). Although this limited the number of participants (to less than 100), the researchers could closely observe their behaviors and gain some insights concerning human-computer interaction. Some papers also conducted formative studies to understand human behaviors before designing the system and the main experiments (Kulesza et al., 2010).

By contrast, some used a crowdsourcing platform, Amazon Mechanical Turk[2] in particular, to collect human feedback for debugging the models. Crowdsourcing (CS) enables researchers to conduct experiments at a large scale; however, the quality of human responses could be varying. So, it is important to ensure some quality control such as specifying required qualifications (Smith-Renner et al., 2020), using multiple annotations per question (Lertvittayakumjorn et al., 2020), having a training phase for participants, and setting up some obvious questions to check whether the participants are paying attention to the tasks, as discussed in (Egelman et al., 2014).

Finally, simulation (SM), without real humans involved but using oracles as human feedback instead, has also been considered. For example, Teso and Kersting (2019) set 20% of input words as relevant using feature selection. These were used to respond to the post-hoc explanations, i.e., top k words selected by LIME. Koh and Liang (2017) simulated mislabeled examples by flipping the labels of a random 10% of the training data. Hence, when the explanation showed suspicious training examples, the true labels could be used to provide feedback. Compared to in-person and crowdsourced experiments, simulation is faster and cheaper. However, its results may not reflect the effectiveness of the framework when deployed with real humans. Naturally, human feedback is sometimes inaccurate and noisy, and humans could also be interrupted or frustrated while providing feedback (Amershi et al., 2014). These factors, which are discussed in detail in the next section, cannot be studied in simulated experiments.

[2] https://www.mturk.com/

3 Research on Human Factors

Although the major goal of EBHD is to improve models, we cannot disregard the effect on human debuggers of the debugging workflow. In this section, we compile findings concerning how explanations and feedback could affect the humans, discussed along five dimensions: model understanding, willingness, trust, frustration, and expectation. Although some of the findings were not derived in NLP settings, we believe that they are generalizable and worth discussing in the context of EBHD.

3.1 Model Understanding

So far, we have used explanations as means to help humans understand models and conduct informed debugging. Hence, it is important to verify, at least preliminarily, that the explanations to be used can accurately create mental models in human debuggers, leading to successful debugging.

Existing studies have found that some explanation forms or methods are more conducive to developing model understanding in humans than others. Stumpf et al. (2009) found that rule-based and keyword-based explanations were easier to understand than similarity-based explanations (i.e., explaining by similar examples in the training data). Also, they found that some users did not understand why the absence of some words could make the model become more certain about its predictions. Lim et al. (2009) found that explaining why the system behaved and did not behave in a certain way resulted in good user's understanding in the system, though the former way of explanation (why) was more effective than the latter (why not). Cheng et al. (2019) reported that interactive explanations could improve users' comprehension on the model better than static explanations, although the interactive way took more time. In addition, they found that revealing inner workings of the model could further help understanding; however, it introduced additional cognitive workload that might reduce people's confidence.

3.2 Willingness

We would like humans to provide feedback for improving models, but do humans naturally want to? Prior to the emerging of EBHD, studies found that humans are not willing to be constantly asked about labels of examples as if they were just simple oracles (Cakmak et al., 2010; Guillory and Bilmes, 2011). Rather, they want to provide more than just

data labels after being given explanations (Amershi et al., 2014; Smith-Renner et al., 2020). By collecting free-form feedback from users, Stumpf et al. (2009) and Ghai et al. (2021) discovered various feedback types. The most prominent ones include removing-adding features (words), tuning weights, and leveraging feature combinations. Stumpf et al. (2009) further analyzed categories of background knowledge underlying the feedback and found, in their experiment, that it was mainly based on commonsense knowledge and English language knowledge. Such knowledge may not be obvious to the model if humans are able to provide only labels. This agrees with some participants, in (Smith-Renner et al., 2020), who described their feedback as inadequate when they could only confirm or correct predicted labels.

Although human feedback beyond labels contains helpful information, it is naturally neither complete nor precise. Ghai et al. (2021) observed that human feedback usually focuses on a few features that are most different from human expectation, ignoring the others. Also, they found that humans, especially lay persons, are not good at correcting model explanations quantitatively (e.g., adjusting weights). This is consistent with the findings of Miller (2019) that human explanations are selective (in a biased way) and rarely refer to probabilities but express causal relationships instead.

3.3 Trust

It has been discussed widely that explanations engender human trust in AI systems (Pu and Chen, 2006; Lipton, 2018; Toreini et al., 2020). This trust may be misplaced at times. Indeed, showing more detailed explanations can cause users to over rely on the system, leading to misuse where users agree with incorrect system predictions (Stumpf et al., 2016). Moreover, some users may over trust the explanations (without fully understanding them) only because the tools generating them are publicly available, widely used, and showing appealing visualizations (Kaur et al., 2020).

However, recent research reported that explanations do not necessarily increase trust and reliance. Cheng et al. (2019) found that, even though explanations help users comprehend systems, they cannot increase human trust in using the systems in high-stakes applications involving lots of qualitative factors, such as graduate school admissions. Smith-Renner et al. (2020) reported that explanations of low-quality models decrease trust and system acceptance as they reveal model weaknesses to the users. This goes along with a perspective by Zhang et al. (2020) that explanations should help calibrate user perceptions to the model quality, signaling whether the users should trust or distrust the AI. Although, in some cases, explanations successfully warned users of faulty models (Ribeiro et al., 2016), this is not easy when the model flaws are not obvious (Zhang et al., 2020; Lertvittayakumjorn and Toni, 2019).

Besides explanations, the effect of feedback on human trust is quite inconclusive according to some (but fewer) studies. On one hand, Smith-Renner et al. (2020) found that, after humans see explanations of low-quality models and lose their trust, the ability to provide feedback makes human trust and acceptance rally, remedying the situation. In contrast, Honeycutt et al. (2020) reported that providing feedback decreases human trust in the system as well as their perception of system accuracy no matter whether the system truly improves after being updated or not. As the effects of explanations and feedback on end users' trust are mixed, it may be preferable to ask ML engineers or domain experts to be debuggers in the workflow, unless feedback can be collected from end users implicitly as suggested in (Honeycutt et al., 2020).

3.4 Frustration

Working with explanations can cause frustration in some situations. Following the discussion on trust, explanations of poor models increase frustration (as they reveal model flaws), whereas the ability to provide feedback reduces frustration. Hence, the most frustrating condition is showing explanations to the users without allowing them to give feedback (Smith-Renner et al., 2020).

Another cause of frustration is the risk of detailed explanations slowing down and overloading users (Narayanan et al., 2018). This is especially a crucial issue for inherently interpretable models where all the internal workings can be exposed to the users. Though presenting all details as explanations is comprehensive and faithful, it could create barriers for lay users (Gershon, 1998). In fact, even ML experts may feel frustrated if they need to understand a decision tree with depth of ten or more. Poursabzi-Sangdeh et al. (2018) found that showing all the model internals undermined users' ability to detect flaws in the model, likely

due to information overload. So, they suggested that model internals should be revealed only when the users request to see them.

3.5 Expectation

Smith-Renner et al. (2020) observed that some participants expected the model to improve over time regardless of whether they got explanations or gave feedback. EBHD frameworks should manage these expectations properly. For instance, the system should report changes or improvements to users after the model gets updated. It would be better if the changes can be seen incrementally in real time (Kulesza et al., 2015).

3.6 Summary

Based on the findings on human factors reviewed in this section, we summarize suggestions for effective EBHD as follows.

Human debuggers. Buggy models usually lead to implausible explanations, adversely affecting human trust in the system. Also, it is not yet clear whether giving feedback increases or decreases human trust. So, where possible, human debuggers should be ML engineers or domain experts rather than end users, unless feedback can be collected from end users implicitly or without letting them know that the explanations are of the system.

Explanations. EBHD should avoid using forms of explanations which are difficult to understand, such as similar training examples and absence of evidence in inputs, unless the humans are already trained to interpret them. Also, too much information should be avoided as it could overload the humans; instead, humans should be allowed to request more information if they are interested, e.g., by using interactive explanations (Dejl et al., 2021).

Feedback. Given that human feedback is not always complete, correct, or accurate, EBHD should use it with care, e.g., by relying on collective feedback rather than individual feedback and allowing human debuggers to verify and modify their feedback before applying it to update the model.

Update. Humans, especially lay people, usually expect the model to improve over time after they give feedback. So, the system should display improvements after the model gets updated. Where possible, showing the changes incrementally to the debuggers in real time is preferred, as they can check if their feedback works as expected or not.

4 Open Problems

This section lists potential research directions and open problems for EBHD of NLP models.

4.1 Beyond English Text Classification

All works in Table 1 conducted experiments only on English datasets. We acknowledge that qualitatively analyzing explanations and feedback in languages at which one is not fluent is not easy, not to mention recruiting human subjects who know the languages. However, we hope that, with more multilingual data publicly available (Wolf et al., 2020) and growing awareness in the NLP community (Bender, 2019), there will be more EBHD studies targeting other languages in the near future.

Also, most existing EBHD works target text classifiers. It would be interesting to explore EBHD for other NLP tasks such as reading comprehension, question answering, and natural language inference (NLI), to see whether existing techniques still work effectively. Shifting to other tasks requires an understanding of specific bug characteristics in those tasks. For instance, unlike bugs in text classification which are usually due to word artifacts, bugs in NLI concern syntactic heuristics between premises and hypotheses (McCoy et al., 2019). Thus, giving human feedback at word level may not be helpful, and more advanced methods may be needed.

4.2 Tackling More Challenging Bugs

Lakkaraju et al. (2020) remarked that the evaluation setup of existing EBHD works is often too easy or unrealistic. For example, bugs are obvious artifacts which could be removed using simple text pre-processing. So, it is not clear how powerful such EBHD frameworks are when dealing with real-world bugs. If bugs are not dominant and happen less often, global explanations may be too coarse-grained to capture them whereas many local explanations may be needed to spot a few appearances of the bugs, leading to inefficiency. This is consistent with the findings of Smith-Renner et al. (2020) that feedback results in minor improvements when the model is already reasonably good.

Other open problems, whose solutions may help deal with challenging bugs, include the following.

- Different people could give different feedback for the same explanation. As raised by Ghai et al. (2021), how can we integrate their feedback to get robust signals for model update? How should we deal with conflicts among

feedback and training examples, e.g., of the kind identified by Carstens and Toni (2017)?

- Confirming or removing what the model has learned is easier than injecting, into the model, new knowledge (which may not even be apparent in the explanations). How can we use human feedback to inject new knowledge, especially when the model is not transparent?

- EBHD techniques have been proposed for tabular data and image data, e.g., as in (Shao et al., 2020; Ghai et al., 2021; Popordanoska et al., 2020). Can we adapt or transfer them across modalities to deal with NLP tasks?

4.3 Analyzing and Enhancing Efficiency

The major focus of most works in Table 1 is the improved correctness of the model after debugging (by expecting a higher F1 or a lower bias, for example). However, only few of them discuss efficiency of the proposed frameworks (Kulesza et al., 2015; Koh and Liang, 2017; Yao et al., 2021). In general, we can analyze the efficiency of an EBHD framework by looking at the efficiency of each main step in Figure 1. Step 1 generates the explanations, so its efficiency depends on the explanation method used and the number of local explanations needed. Step 2 lets humans give feedback, so its efficiency concerns the amount of time they spend to understand the explanations and to produce the feedback. Step 3 updates the model using the feedback, so its efficiency relates to the time used for processing the feedback and retraining the model (if needed). Existing works mainly reported efficiency of step 2. For instance, Kulesza et al. (2015) compared the improved F1 of EBHD with F1 of instance labelling given the same amount of time for humans to perform the task. Conversely, Yao et al. (2021) compared the time humans need to do EBHD versus instance labelling in order to achieve the equivalent degree of correctness improvement.

None of the works we survey considered the efficiency of the three steps altogether. In fact, the efficiency of step 1 and 3 is important especially for black box models where the cost of post-hoc explanation generation and model retraining is not negligible. It is even crucial for iterative or responsive EBHD. So, analyzing and enhancing efficiency of EBHD frameworks (for both machine and human sides) require further research.

4.4 Reliable Comparison across Papers

User studies are naturally difficult to replicate because they are inevitably affected by choices of user interfaces, phrasing, population, incentives, etc. (Lakkaraju et al., 2020). Additionally, research in ML rarely adopts practices from the human-computer interaction community (Abdul et al., 2018), limiting the possibility to compare across studies. Hence, most existing works only consider model performance before and after debugging or compare the results among several configurations of a single proposed framework. This leads to little knowledge in the community about which explanation types or feedback mechanisms are more effective than others across several settings. Therefore, one promising research direction would be proposing a standard setup or a benchmark for evaluating and comparing EBHD frameworks reliably across different settings.

4.5 Towards Deployment

So far, EBHD research has not been deployed in applications, probably because of its difficulty to set up the debugging aspects outside a research environment. One way to promote adoption of EBHD is to integrate EBHD frameworks into available visualization systems such as the Language Interpretability Tool (LIT) (Tenney et al., 2020), allowing users to provide feedback to the model after seeing explanations and supporting experimentation. Also, to move towards deployment, it is important to follow human-AI interaction guidelines (Amershi et al., 2019) and evaluate EBHD with potential end users, not just via simulation or crowdsourcing, since human factors play an important role in real situations (Amershi et al., 2014).

5 Conclusion

We presented a general framework of explanation-based human debugging (EBHD) of NLP models and analyzed existing works in relation to the components of this framework to illustrate the state-of-the-art in the field. Furthermore, we summarized findings on human factors with respect to EBHD, suggested design practices accordingly, and identified open problems for future studies. As EBHD is still an ongoing research topic, we hope that our survey will be helpful for guiding interested researchers and for examining future EBHD papers.

References

Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–18.

Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. 2020. Debugging tests for model explanations. In *Advances in Neural Information Processing Systems*.

Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120.

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. 2019. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*.

Emily Bender. 2019. The #benderrule: On naming the languages we study and why it matters. *The Gradient*.

Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. 2020. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657.

Gabriel Cadamuro, Ran Gilad-Bachrach, and Xiaojin Zhu. 2016. Debugging machine learning models. In *ICML Workshop on Reliable Machine Learning in the Wild*.

Maya Cakmak, Crystal Chao, and Andrea L Thomaz. 2010. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: natural language inference with natural language explanations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9560–9572.

Lucas Carstens and Francesca Toni. 2017. Using argumentation to improve classification in natural language problems. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1–23.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730.

Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F Maxwell Harper, and Haiyi Zhu. 2019. Explaining decision-making algorithms through ui: Strategies to help non-expert stakeholders. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–12.

Minseok Cho, Reinald Kim Amplayo, Seung-won Hwang, and Jonghyuck Park. 2018. Adversarial tableqa: Attention supervision for question answering on tables. In *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 391–406. PMLR.

Minseok Cho, Gyeongbok Lee, and Seung-won Hwang. 2019. Explanatory and actionable de-

bugging for machine learning: A tableqa demonstration. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1333–1336.

Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-adapted interaction*, 18(5):455.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 120–128, New York, NY, USA. Association for Computing Machinery.

Adam Dejl, Peter He, Pranav Mangal, Hasan Mohsin, Bogdan Surdu, Eduard Voinea, Emanuele Albini, Piyawat Lertvittayakumjorn, Antonio Rago, and Francesca Toni. 2021. Argflow: A toolkit for deep argumentative explanations for neural networks. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1761–1763.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Serge Egelman, Ed H Chi, and Steven Dow. 2014. Crowdsourcing in hci research. In *Ways of Knowing in HCI*, pages 267–289. Springer.

Rebecca Fiebrink, Dan Trueman, and Perry R. Cook. 2009. A metainstrument for interactive, on-the-fly machine learning. In *In Proc. NIME*.

Nahum Gershon. 1998. Visualization of an imperfect world. *IEEE Computer Graphics and Applications*, 18(4):43–45.

Bhavya Ghai, Q Vera Liao, Yunfeng Zhang, Rachel Bellamy, and Klaus Mueller. 2021. Explainable active learning (xal) toward ai explanations as interfaces for machine teachers. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–28.

Filip Graliński, Anna Wróblewska, Tomasz Stanisławek, Kamil Grabowski, and Tomasz Górecki. 2019. GEval: Tool for debugging NLP datasets and models. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 254–262, Florence, Italy. Association for Computational Linguistics.

Andrew Guillory and Jeff Bilmes. 2011. Simultaneous learning and covering with adversarial noise. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 369–376, Madison, WI, USA. Omnipress.

Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. 2020. Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Donald Honeycutt, Mahsan Nourani, and Eric Ragan. 2020. Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions of model accuracy. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 63–72.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2020. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. *arXiv preprint arXiv:2010.07487*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*.

Daniel Kang, Deepti Raghavan, Peter Bailis, and Matei Zaharia. 2018. Model assertions for debugging machine learning. In *NeurIPS MLSys Workshop*.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. 2020. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Sung Wook Kim, Iljeok Kim, Jonghwan Lee, and Seungchul Lee. 2021. Knowledge integration into deep learning in dynamical systems: an overview and taxonomy. *Journal of Mechanical Science and Technology*, pages 1–12.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 5686–5697.

Sanjay Krishnan and Eugene Wu. 2017. Palm: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, pages 1–6.

Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the*

20th international conference on intelligent user interfaces, pages 126–137.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Weng-Keen Wong, Yann Riche, Travis Moore, Ian Oberst, Amber Shinsel, and Kevin McIntosh. 2010. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 41–48. IEEE.

Todd Kulesza, Weng-Keen Wong, Simone Stumpf, Stephen Perona, Rachel White, Margaret M Burnett, Ian Oberst, and Andrew J Ko. 2009. Fixing the program my computer learned: Barriers for end users, challenges for the machine. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 187–196.

Vivian Lai, Han Liu, and Chenhao Tan. 2020. " why is' chicago'deceptive?" towards building model-driven tutorials for humans. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Vivian Lai and Chenhao Tan. 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 29–38.

Himabindu Lakkaraju, Julius Adebayo, and Sameer Singh. 2020. Explaining machine learning predictions: State-of-the-art, challenges, and opportunities. NeurIPS 2020 Tutorial.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Piyawat Lertvittayakumjorn, Lucia Specia, and Francesca Toni. 2020. FIND: Human-in-the-Loop Debugging Deep Text Classifiers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 332–348, Online. Association for Computational Linguistics.

Piyawat Lertvittayakumjorn and Francesca Toni. 2019. Human-grounded evaluations of explanation methods for text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5195–5205, Hong Kong, China. Association for Computational Linguistics.

Brian Y Lim, Anind K Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2119–2128.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Raoni Lourenço, Juliana Freire, and Dennis Shasha. 2019. Debugging machine learning pipelines. In *Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning*, pages 1–10.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Brad A Myers, David A Weitzman, Andrew J Ko, and Duen H Chau. 2006. Answering why and why not questions in user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 397–406.

Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*.

Devi Parikh and C Zitnick. 2011. Human-debugging of machines. *NIPS WCSSWC*, 2(7):3.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium. Association for Computational Linguistics.

Teodora Popordanoska, Mohit Kumar, and Stefano Teso. 2020. Machine guides, human supervises: Interactive learning with global explanations. *arXiv preprint arXiv:2009.09723*.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*.

Pearl Pu and Li Chen. 2006. Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Daniel Selsam, Percy Liang, and David L Dill. 2017. Developing bug-free machine learning systems with formal mathematics. In *International Conference on Machine Learning*, pages 3047–3056. PMLR.

Xiaoting Shao, Tjitze Rienstra, Matthias Thimm, and Kristian Kersting. 2020. Towards understanding and arguing with classifiers: Recent progress. *Datenbank-Spektrum*, 20(2):171–180.

Alison Smith-Renner, Ron Fan, Melissa Birchfield, Tongshuang Wu, Jordan Boyd-Graber, Daniel S Weld, and Leah Findlater. 2020. No explainability without accountability: An empirical study of explanations and feedback in interactive ml. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Simone Stumpf, Adrian Bussone, and Dympna O'sullivan. 2016. Explanations considered harmful? user interactions with machine learning systems. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*.

Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International journal of human-computer studies*, 67(8):639–662.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The

language interpretability tool: Extensible, interactive visualizations and analysis for NLP models.

Stefano Teso and Kristian Kersting. 2019. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 239–245.

Ehsan Toreini, Mhairi Aitken, Kovila Coopamootoo, Karen Elliott, Carlos Gonzalez Zelaya, and Aad van Moorsel. 2020. The relationship between trust in ai and trustworthy machine learning technologies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 272–283.

Zijie J Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey. *arXiv preprint arXiv:2103.04044*.

Thomas Wolf, Quentin Lhoest, Patrick von Platen, Yacine Jernite, Mariama Drame, Julien Plu, Julien Chaumond, Clement Delangue, Clara Ma, Abhishek Thakur, Suraj Patil, Joe Davison, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angie McMillan-Major, Simon Brandeis, Sylvain Gugger, François Lagunas, Lysandre Debut, Morgan Funtowicz, Anthony Moi, Sasha Rush, Philipp Schmidd, Pierric Cistac, Victor Muštar, Jeff Boudier, and Anna Tordjmann. 2020. Datasets. *GitHub. Note: https://github.com/huggingface/datasets*, 1.

Tongshuang Wu, Daniel S Weld, and Jeffrey Heer. 2019. Local decision pitfalls in interactive machine learning: An investigation into feature selection in sentiment analysis. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 26(4):1–27.

Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining neural networks with compositional explanations. *arXiv preprint arXiv:2103.10415*.

Roozbeh Yousefzadeh and Dianne P O'Leary. 2019. Debugging trained machine learning models using flip points. In *ICLR 2019 Debugging Machine Learning Models Workshop*.

Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305.