Are Good Explainers Secretly Human-in-the-Loop Active Learners?

Emma Thuong Nguyen [*1] **Abhishek Ghose** [*1]

Abstract

Explainable AI (XAI) techniques have become popular for multiple use-cases in the past few years. Here we consider its use in studying model predictions to gather additional training data. We argue that this is equivalent to Active Learning, where the query strategy involves a human-in-the-loop. We provide a mathematical approximation for the role of the human, and present a general formalization of the end-to-end workflow. This enables us to rigorously compare this use with standard Active Learning algorithms, while allowing for extensions to the workflow. An added benefit is that their utility can be assessed via simulation instead of conducting expensive user-studies. We also present some initial promising results.

1. Introduction

Keeping in pace with the popularity of Machine Learning (ML), the past few years has seen a surge in the desire to understand how a model makes decisions. In certain domains, such as healthcare and law enforcement, such transparency is critical in acquiring the trust of its users. In others, it serves as a way to understand potential shortcomings of a system, e.g., if a system has overfit the data. This has led to accelerated research in the area of *Explainable AI (XAI)*, which studies explaining of predictions from a given model, e.g. *LIME* (Ribeiro et al., 2016), *SHAP* (Lundberg & Lee, 2017), *DeepLIFT* (Shrikumar et al., 2017).

Here we consider the latter use of XAI: that of improving an ML classifier. We specifically look at the use of explanations to identify data that we deem "interesting" in some way, that is then used to further train our model. As an example, consider the workflow shown in Figure 1, describing the following sequence of events:

1. An ML practitioner trains a supervised classifier on an initial labeled dataset - (X_{orig}, Y_{orig}) - and deploys it (in Figure 1, this is shown by `A`). The model is referred to as *Model v1*.
2. Users of this system interact with it, and in the process, incrementally generate (unlabeled) data, X_{inc}. Shown by `B` in Figure 1.
3. The ML practitioner periodically inspects the system for correctness. She samples from X_{inc} and uses an explainer to review the model's decision process. Shown by `C` in Figure 1 - note that the model is required as an input to the explainer.

   Some explanations might indicate unintended behavior of the model. For example, both these reviews may be classified as positive, where the explainer has underlined the words that most influenced the classifier's decision:

   - I love the food here!
   - I gave them a 1-star rating - that's how much I like the food here.

   Of course, the second review is sarcastic, and should be identified as negative.
4. The ML practitioner decides to sample more such examples from X_{inc} (shown by `D` in Figure 1), and then has them labeled by human annotators (shown by `E`). This new dataset[1] is denoted by (X_{new}, Y_{new}), and is used to further train the model to obtain *Model v2* (shown by `F`).

This process is repeated multiple times to generate improved versions of models. Figure 1 shows one such iteration. While this process seems intuitively appealing, we make rigorous the following aspects:

1. First, we claim that this process essentially is *Active Learning* (Settles, 2009) that involves a human-in-the-loop (Section 2).

[*]Equal contribution [1][24]7.ai, California, USA. Correspondence to: Emma Thuong Nguyen <emma.nguyen@247.ai>, Abhishek Ghose <abhishek.ghose@247.ai>.

AI & HCI Workshop at the 40^{th} International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

[1]This new dataset may be seen to contain only the newly identified instances if the model may be incrementally trained, or a combination of the original data and the new instances, if the model needs to be trained from scratch. We will adopt the latter convention here since its universal, i.e., not all models support incremental training.

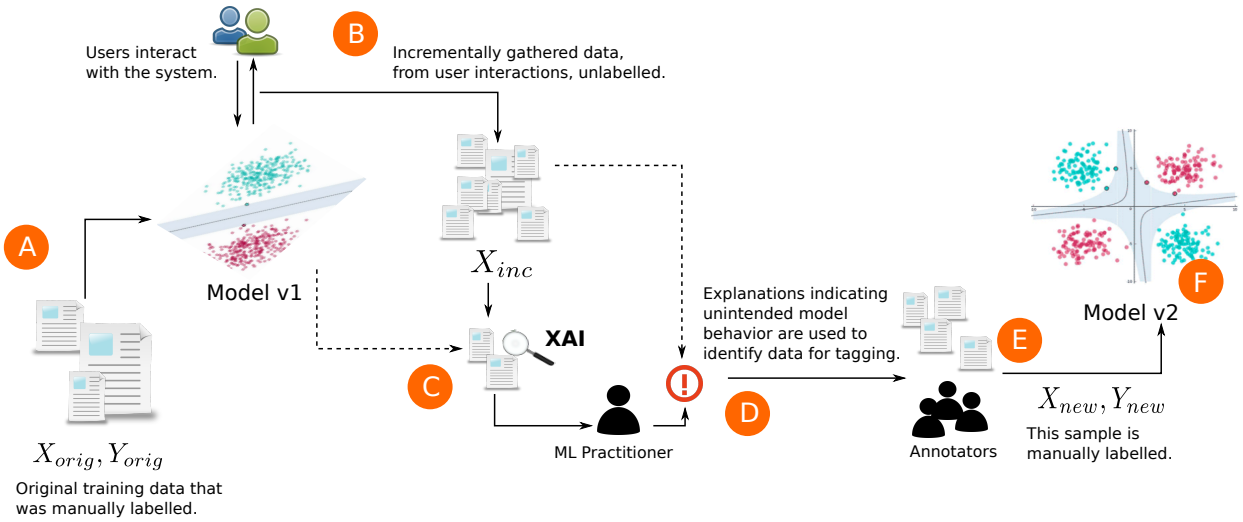

Figure 1. Workflow representing use of explanations to identify data to retrain a model. This shows one iteration of such a workflow, where we start with *Model v1* and create a more accurate model *Model v2*, based on sampling new training instances from a data pool, X_{inc}. Please see Section 1 for details. As we shown in Equation 13 in Section 3, many of these steps may be distilled into a single mathematical expression.

2. Then, we mathematically formulate the workflow, which makes it convenient to (a) quantify its utility, and (b) compare with other AL techniques such as *margin-based sampling* (Scheffer et al., 2001) (Section 3).

   An added benefit is such workflows may be evaluated via simulation bypassing the need for conducting time-consuming or expensive user studies.

Our primary interest is text classification, but much of the discussion here applies to other forms of data as well.

Related Work: Based on a thorough search of the relevant literature, we believe this is the first study that casts human-in-the-loop data selection based on explanations as an AL query strategy. Other intersections of AL and XAI have been studied however, e.g., Liu et al. (2021) uses certain properties of local explanations to determine instance-informativeness for querying, Ghai et al. (2021) studies the benefits of annotating queried instances with explanations to obtain rich feedback from a human labeler.

2. XAI-based Data Selection is Active Learning

In many situations, while abundant unlabeled data is available, labeled data may be hard to procure, e.g., when manual annotation by experts is required. In such cases, one needs to explicitly account for the *label acquisition cost*. The *Active Learning (AL)* family of techniques solves for this problem in the following way:

1. An initial model is built by acquiring labels for a small batch of data. This batch may be randomly selected from the unlabeled pool of data.

2. The model is then iteratively improved by *strategically* selecting data (from the unlabeled pool) to be annotated. This data is then used to further train the model. Such a strategy or *query strategy*[2] picks instances that have the greatest influence on the model's accuracy. Some popular query strategies are *entropy sampling* and *maximum margin sampling*.

Informally, the query strategy is a mechanism to identify maximally useful instances given (a) the current model, and (b) an unlabeled pool of data. Referring to Figure 1, we observe the following components effectively form a query strategy:

- The explainer used to detect surprising patterns in model predictions (shown by C in Figure 1).
- The process of using the explanations to solicit further instances from the unlabeled pool X_{inc} (D in Figure 1).

Specifically, this is the *batch* AL setting, where batches of data are iteratively identified, labeled and used to train the

[2]So called because it is used to query instances from the unlabeled pool.

model, e.g., *BatchBALD* (Kirsch et al., 2019). AL may be used in various other settings as well, such as *stream-based* - see Settles (2009) for an overview.

3. Mathematical Formulation

How do we compare this form of AL with standard AL techniques? Clearly, a challenge is that because there is a human-in-the-loop - the ML practitioner - this workflow needs to be tested with expensive or time consuming user-studies. In this section, we try to eliminate this roadblock by (1) providing a reasonable approximation for the task of the ML practitioner, and (2) offering a concise representation for the overall workflow. This makes it possible to efficiently simulate the workflow from Figure 1.

We introduce some notation first:

1. We will denote the number of instances in the collection X_a by N_a. We will also assume that our data resides in d dimensions, i.e., $X_{orig} \in \mathbb{R}^{N_{orig} \times d}$, $X_{inc} \in \mathbb{R}^{N_{inc} \times d}$ and $X_{new} \in \mathbb{R}^{N_{new} \times d}$.

2. We will assume explanations are produced in d' dimensions. The case of $d \neq d'$ is common for text explainers where the text input that a model sees maybe in form of *n-grams* or *Byte Pair Encoding (BPE)* (Sennrich et al., 2016) vectors, whereas the explanation might be in an "interpretable" space such as presence/absence of words. For tabular data $d = d'$. We'll use the "$'$" superscript to denote data in the explanation space, e.g., $X'_{orig} \in \mathbb{R}^{N_{orig} \times d'}$.

3. Models *Model v1* and *Model v2* are denoted by the function f, parameterized by Ψ_1 and Ψ_2 respectively[3]. As examples, Ψ may be coefficients in *Logistic Regression* or weights in a *Neural Network*.

4. The explainer is denoted by the function $E(x; \theta, \Psi)$, where $x \in \mathbb{R}^d$ is an instance for which an explanation for its prediction by model Ψ is sought. The explainer itself has parameters θ, such as the number of features to be used in explanations (Ribeiro et al., 2016).

5. The explanation is a vector of weights $q \in \mathbb{R}^{d'}$ that explains the input x in the explanation space, i.e., it applies to $x' \in \mathbb{R}^{d'}$. Intuitively, these weights indicate the importance of the corresponding feature.

   While this specific format for explanations is an assumption, it is common (Ribeiro et al., 2016; Lundberg & Lee, 2017; Kim et al., 2020; Slack et al., 2021) and allows our formulation to be broadly applicable.

6. Finally, we account for two practical constraints in our setup:

   (a) $\mathcal{B}_E$, explanation budget: the number of instances whose explanations an ML practitioner might manually study.

   (b) $\mathcal{B}_L$, labeling budget: the number of instances that annotators can label within one iteration. This is equivalent to the *batch size* in AL.

   Typically, for real-world systems, $\mathcal{B}_E < \mathcal{B}_L < N_{inc}$. Some representative numbers are: $\mathcal{B}_E$ is in the order of hundreds, $\mathcal{B}_L$ is in the order of hundreds to thousands (depending on the labeling cost, e.g., skill required, number of annotators), and N_{inc} may be arbitrarily large, potentially running into millions of instances.

[3] As mentioned earlier, we discuss only one iteration of model improvement, but this discussion applies to the general case of learning Ψ_{i+1}, given Ψ_i.

3.1. Task Formulation

Given the above notation, we now revisit the workflow from Figure 1:

1. *Step C in Figure 1*: Explanations $E(x_i; \theta, \Psi_1)$ are sought for instances $x_i \in X_s$, where $X_s \subseteq X_{inc}$ is a set of instances randomly selected from X_{inc}, such that its size N_s does not exceed the explanation budget $\mathcal{B}_E$. For each instance $x_i \in \mathbb{R}^d$, an explanation weight vector $q_i \in \mathbb{R}^{d'}$ is produced.

2. *Step D, representing unintended model behavior*: Based on studying the explanations, the ML practitioner identifies instances in X_s that indicate model behavior that is either unintended or in some sense, surprising. An example is that different labels are predicted for a pair of instances that are either similar or produce similar explanations. Intuitively, this might mean the model requires more such instances to confidently tell them apart.

   Recall that the practitioner's goal is to select instances from X_{inc} similar to the ones that participate in such pairs in X_s. And, since she can select only up to $\mathcal{B}_L$ instances, we want to favour instances in X_s that participate in a large number of such pairs.

   We represent this in the following manner (note that all x appearing below belong to X_s):

   - Let matrix $A \in \mathbb{R}^{N_s \times N_s}$ represent similarity between instances - either in terms of the vectors themselves, or their explanations. We combine them in the following way:

   $$A_{ij} = (q_i \odot x'_i) \cdot (q_j \odot x'_j)^T \qquad (1)$$

   The "$\odot$" symbol represents the *element-wise* product and the "$\cdot$" symbol denotes the *dot product*.

- Let $B \in \mathbb{R}^{N_s \times N_s}$ represent whether predicted labels are identical:

$$B_{ij} = \begin{cases} 1 & f(x_i, \Psi_1) \neq f(x_j, \Psi_1) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

  Note that pairs of instances are assigned a value of 1 when the predicted labels are *not* identical.
- Finally, given the identity vector $\mathbb{1}_{N_s} \in \mathbb{R}^{N_s \times 1}$, we[4] define $C \in \mathbb{R}^{N_s}$:

$$C = (A \odot B)\mathbb{1}_{N_s} \quad (3)$$

  Consider the values for $A \odot B$:
  (a) +ve values indicate pairs of instances with different predictions, i.e., $B_{ij} = 1$, but similar explanations, i.e., $A_{ij} > 0$ (preferred for retrieval).
  (b) -ve values indicate different predictions, i.e., $B_{ij} = 1$, but also different explanations, i.e., $A_{ij} < 0$ (not preferred).
  (c) 0 values either indicate same predictions, i.e., $B_{ij} = 0$ or different explanations , i.e., $A_{ij} = 0$ (not preferred).

  C_i provides a row-wise sum for instances x_i in $A \odot B$, quantifying the extent to which they are preferred during retrieval.

  As an examples, consider for $N_s = 3$:

$$A = \begin{bmatrix} 1 & 0.7 & 0.8 \\ 0.7 & 1 & 0.3 \\ 0.8 & 0.3 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (4)$$

$$A \odot B = \begin{bmatrix} 0 & 0.7 & 0.8 \\ 0.7 & 0 & 0 \\ 0.8 & 0 & 0 \end{bmatrix} \quad (5)$$

$$C = (A \odot B)\mathbb{1}_{N_s} = \begin{bmatrix} 1.5 \\ 0.7 \\ 0.8 \end{bmatrix} \quad (6)$$

  Here, as per B, x_1 and x_2 have the same predicted label, which is different from that of x_0. As per A, x_0 also has high explanation similarities to both x_1 and x_2. The desirability of x_0 is reflected in its high value in C. Intuitively, C indicates preference weights for instances in X_s, to be used as queries for retrieval.

3. *Step D (continued), retrieving instances from X_{inc}*: We now select $\mathcal{B}_\mathcal{L}$ instances from X_{inc} based on C. This is a human-in-the-loop activity, which we assume may be approximated with similarities via dot products. We detail this below.

[4]The identity vector has a subscript that denotes its length. Also, we will assume, vectors are *column* vectors.

We compute the following score matrix $S \in \mathbb{R}^{N_{inc} \times N_s}$:

$$S = X'_{inc}(C\mathbb{I}^T_{d'} \odot X'_s)^T \quad (7)$$

Note that X'_{inc} and X'_s are representations in the explanation space. $C\mathbb{I}^T_{d'} \odot X'_s$ multiplies each vector in X_s with the corresponding weight in C. Finally, S computes the similarity, i.e., dot-product, between vectors in X_{inc} and these weighted vectors from X'_s.

To continue with our example, let's consider the following X'_s with $d' = 2$:

$$X'_s = \begin{bmatrix} 0.7 & 0.2 \\ 0.34 & 1.15 \\ -0.1 & 3 \end{bmatrix}, \quad (8)$$

Then,

$$C\mathbb{I}^T_{d'} = \begin{bmatrix} 1.5 \\ 0.7 \\ 0.8 \end{bmatrix} \begin{bmatrix} 11 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 \\ 0.7 & 0.7 \\ 0.8 & 0.8 \end{bmatrix} \quad (9)$$

$$C\mathbb{I}^T_{d'} \odot X'_s = \begin{bmatrix} 1.5 \times 0.7 & 1.5 \times 0.2 \\ 0.7 \times 0.34 & 0.7 \times 1.15 \\ 0.8 \times -0.1 & 0.8 \times 3 \end{bmatrix}, \quad (10)$$

S is a symmetric matrix, where S_{ij} denotes the similarity between $x'_i \in X'_{inc}$ and $x'_j \in X'_s$ (or indirectly, x_i and x_j), accounting for the preferences encoded by C.

To obtain the overall retrieval desirability for an instance in X'_{inc}, we compute its row-wise sum. We define the retrieval weights $W \in \mathbb{R}^{N_{inc}}$ as:

$$W = S\mathbb{1}_{N_s} \quad (11)$$

We select the top $\mathcal{B}_L$ instances from X_{inc} based on W. We will refer to these instances as $X_{top} \in \mathbb{R}^{\mathcal{B}_L \times d}$.

4. *Steps E and F*: We obtain labels $Y_{top} \in \mathbb{R}^{\mathcal{B}_L}$ corresponding to X_{top} via human annotation, and construct the following new dataset:

$$X_{new} = \begin{bmatrix} X_{orig} \\ X_{top} \end{bmatrix}, X_{new} \in \mathbb{R}^{(N_{orig}+\mathcal{B}_L) \times d}$$
$$Y_{new} = \begin{bmatrix} Y_{orig} \\ Y_{top} \end{bmatrix}, Y_{new} \in \mathbb{R}^{(N_{orig}+\mathcal{B}_L)} \quad (12)$$

(X_{new}, Y_{new}) is used to retrain f obtaining new parameters Ψ_2.

These steps can be condensed into a single expression - given matrices A (this makes use of explanations) and B, the top-$\mathcal{B}_L$ instances from X_{inc} based on these weights are picked:

$$W = X'_{inc}(((A \odot B)\mathbb{1}_{N_s})\mathbb{I}^T_{d'} \odot X'_s)^T\mathbb{1}_{N_s} \quad (13)$$

3.2. Objective

We want to minimize the generalization loss of f. In an AL setting the only labeled data available is (X_{new}, Y_{new}) (Equation 12) and therefore, a validation set must be sampled from it. To keep the notation simple, we use $\mathcal{L}_v$ to denote generalization loss, i.e., loss on a validation set, and our objective as:

$$\min_{\theta,\Psi} \mathcal{L}_v(X_{orig}, Y_{orig}, X_{inc}, \theta, \Psi) \tag{14}$$

Note here that we optimize for the explanation parameters θ as well to refine them to be helpful in the data selection process, i.e., construction of (X_{new}, Y_{new}).

3.3. Metrics

To measure competitiveness against standard AL approaches, we hold out a labeled test set (X_{test}, Y_{test}) that is large enough to reflect the true distribution. Such a dataset is not available in real-world AL setups, and is used here to measure the true accuracy of a model. We report model accuracy scores on this dataset, at various iterations of data being sampled from X_{inc}.

3.4. Extensions

While we looked at at one form of unintended behavior here - different predictions but similar explanations - it is possible to define others. These may be defined based on (X_{orig}, Y_{orig}) as well. Some examples are:

1. For an instance $x_i \in X_{orig}$, the predicted label is incorrect, but the explanations for both predicted label and true label are similar. It is possible to specify this behavior only if the explainer can generate explanations for membership to any class, e.g., LIME, SHAP.

   Intuition: the model is misaligned with respect to its mapping from features to labels.

2. For a pair of instances with different true labels, the explanations for membership to these labels are similar.

   Intuition: the model is unable to strongly discriminate between the two classes for these instances, and might need more similar instances to learn.

These criteria can be easily included in our formulation by appropriately defining matrices A and B.

3.5. Review of Assumptions

In our formulation we make two assumptions:

- The format of explanations as a weight vector. As mentioned earlier, this is indeed a common format, and allows the formulation to be broadly applicable.
- Approximating the human-in-the-loop process with retrieval based on dot-product based similarities. While this is probably reasonable, we require user studies to validate its adequacy.

□

We note that the above formulation is generic and applicable to different kinds of data, e.g., text, images, tabular, different explainers, as well as different models.

4. Experiments

We have begun empirical comparisons to standard AL techniques. While our goal is to cover a diverse set of data, models and explainers, we present initial results on the dataset SST-5 (Socher et al., 2013) using SHAP (Lundberg & Lee, 2017), specifically Partition SHAP, as the explainer and *Support Vector Machine* with linear kernel as our model (the *scikit-learn* (Buitinck et al., 2013) library is used). We use the *F1-macro* score to report accuracy; this metric is used since it accounts for class-wise accuracies even when there is class imbalance. The optimization in Equation 14 is solved using *Bayesian Optimization* (Shahriari et al., 2016), since they enable us to minimize non-differentiable functions. This is an important consideration since we are not guaranteed differentiability, e.g., when using a Decision Tree as our model. We specifically use the *Ray Tune* (Liaw et al., 2018) and *Optuna* (Akiba et al., 2019) libraries. For AL, we use the *modAL* library (Danka & Horvath, 2018).

For reasons of tractability, the joint optimization over θ and Ψ (Equation 14) is decomposed into the following nested optimization:

1. The search space of the explanation parameters θ is explored by the Bayesian Optimizer. The SHAP parameters we varied are maximum number of predictions which model f makes for explaining one instance and maximum number of predictions in one model invocation.

2. The model selection search space is explored by standard cross-validation with grid-search over hyperparameters: in this case, the regularization coefficient C.

Other relevant details:

1. The text representation used is *Universal Sentence Encoding (USE)* (Cer et al., 2018).

2. Experiment settings:

   - $N_{orig} = 100$, $N_{inc} = 2900$, $N_{test} = 2000$.

Figure 2. F1-macro evaluated on (X_{test}, y_{test}) for the dataset SST5. The x-axis describes the number of instances used in each model training iteration. The solid line represents the accuracy for our explanation based AL (Section 3.1). The dotted lines represent results for different popular AL methods: *entropy sampling*, *random sampling*, *maximum margin sampling*. The band around each line indicates 95% confidence intervals across 4 runs.

- Labeling budget (or batch size in AL), $\mathcal{B}_L = 200$, explanation budget $\mathcal{B}_L = 200$.

3. Comparisons against:
   - AL query strategies: entropy-based sampling[5], margin-based sampling[6] (Scheffer et al., 2001). See Settles (2009) for an overview.
   - Baseline: we use a *random* strategy, which selects $\mathcal{B}_L$ instances from X_{inc} uniformly at random with no replacement.

We visualize our results in Figure 2. "AL" or "EXP" in the legend denote whether a strategy comes from standard AL or is based on an explainer. We observe that the explanation based query strategy performs better than other standard AL techniques. It achieves higher scores right at the first few iterations and reaches a plateau in performance. We also note that the AL query strategies are not significantly better than random selection.

5. Conclusions and Future Work

In this short paper, we investigated a specific use of XAI: using it to select model training data. We showed that this is equivalent to performing Active Learning, with a human-in-the-loop as part of the query strategy. Further, we mathematically approximated this workflow, so that it may be conveniently studied and empirically compared to other Active Learning techniques. We presented some initial results; these look promising, and we hope to continue the empirical analyses to definitively establish the utility of XAI in this setup.

Our future work would focus on: (a) validating our approximation for the human-in-the-loop process via user-studies, (b) broadening the scope of this study by using different classifiers, text representations, datasets and AL techniques, especially those recently proposed, e.g., Zhdanov (2019), Cardoso et al. (2017), and (c) exploiting the differentiability of our formulation (Equation 13) to *learn* an AL strategy.

[5]This selects instances with high entropy values over prediction probabilities across classes.

[6]Selects instances that have a small difference between the prediction probabilities of the two most confident classes.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for ma-

chine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Cardoso, T. N., Silva, R. M., Canuto, S., Moro, M. M., and Gonçalves, M. A. Ranked batch-mode active learning. *Information Sciences*, 379:313–337, 2017. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2016.10.037. URL https://www.sciencedirect.com/science/article/pii/S0020025516313949.

Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N. L. U., John, R. S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., hsuan Sung, Y., Strope, B., and Kurzweil, R. Universal sentence encoder. In *In submission to: EMNLP demonstration*, Brussels, Belgium, 2018. URL https://arxiv.org/abs/1803.11175. In submission.

Danka, T. and Horvath, P. modAL: A modular active learning framework for Python. 2018. URL https://github.com/cosmic-cortex/modAL. available on arXiv at https://arxiv.org/abs/1805.00979.

Ghai, B., Liao, Q. V., Zhang, Y., Bellamy, R., and Mueller, K. Explainable active learning (xal): Toward ai explanations as interfaces for machine teachers. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW3), jan 2021. doi: 10.1145/3432934. URL https://doi.org/10.1145/3432934.

Kim, S., Yi, J., Kim, E., and Yoon, S. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3154–3167, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.255. URL https://aclanthology.org/2020.emnlp-main.255.

Kirsch, A., van Amersfoort, J., and Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/95323660ed2124450caaac2c46b5ed90-Paper.pdf.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

Liu, Q., Zhu, Y., Liu, Z., Zhang, Y., and Wu, S. Deep active learning for text classification with diverse interpretations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 3263–3267, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482080. URL https://doi.org/10.1145/3459637.3482080.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Ribeiro, M., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-3020. URL https://aclanthology.org/N16-3020.

Scheffer, T., Decomain, C., and Wrobel, S. Active hidden markov models for information extraction. In Hoffmann, F., Hand, D. J., Adams, N., Fisher, D., and Guimaraes, G. (eds.), *Advances in Intelligent Data Analysis*, pp. 309–318, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44816-7.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Settles, B. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differ-

ences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3145–3153. JMLR.org, 2017.

Slack, D. Z., Hilgard, S., Singh, S., and Lakkaraju, H. Reliable post hoc explanations: Modeling uncertainty in explainability. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=rqfq0CYIekd.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Zhdanov, F. Diverse mini-batch active learning. *ArXiv*, abs/1901.05954, 2019. URL http://arxiv.org/abs/1901.05954.