Human-in-the-Loop Rule Discovery for Micropost Event Detection

Akansha Bhardwaj, Jie Yang, Philippe Cudré-Mauroux

Abstract—Platforms such as Twitter are increasingly being used for real-world event detection. Recent work often leverages event-related keywords for training machine learning based event detection models. These approaches make strong assumptions on the distribution of the relevant microposts containing the keyword – referred to as the expectation – and use it as a posterior regularization parameter during model training. Such approaches are, however, limited by the informativeness of the keywords and by the accuracy of the expectation estimation for keywords. In this work, we introduce a human-in-the-loop approach to jointly discover informative rules for model training while estimating their expectation. Our approach iteratively leverages the crowd to estimate both rule-specific expectation and the disagreement between the crowd and the model in order to discover new rules that are most beneficial for model training. To identify such rules, we introduce a hybrid human-machine workflow that engages human workers in rule discovery through an interactive hypothesis creation and testing interface and leverages automatic methods for suggesting useful rules for human verification. We empirically demonstrate the merits of our approach, on multiple real-world datasets and show that our approach improves the state of the art by a margin of 25.63% in terms of AUC.

Index Terms—Event Detection, Human-in-the-loop AI, Rules in Machine Learning, Interactive Machine Learning

1 Introduction

Microblogging platforms are important sources of information about real-time events happening around the world and are leveraged by many news agencies for the task of event detection. For example, Twitter is a popular microblogging service that has a monthly active user list of 145M daily active users who post about 500M tweets per day[1]. An important task in the field of event detection is detecting events of *predetermined types* [1], such as concerts or controversial events based on microposts matching specific event descriptions. This task has extensive applications ranging from cyber security [2, 3], political elections [4] to public health [5, 6].

Due to the highly ambiguous and inconsistent terms used in microposts, event detection is generally performed using statistical machine learning models to classify the relevance of microposts to a given event type. Training those models often requires a large set of labeled microposts, which is laborious and costly to create. More specifically, though we can collect positive labels (e.g., using targeted hashtags, or event-related date-time information), there is no straightforward way to generate negative labels that are useful for model training. To fill this gap, Ritter et al.(2015) introduced a weakly supervised approach, which uses only positively labeled data, accompanied by unlabeled examples by filtering microposts that contain a certain keyword indicative of the event type under consideration (e.g., 'hack' for cyber security).

• *Akansha Bhardwaj and Philippe Cudré-Mauroux are with the Department of Computer Science, University of Fribourg, Switzerland, E-mail: akansha.bhardwaj, philippe.cudre-mauroux@unifr.ch. Jie Yang (corresponding author) is with the Web Information Systems group, Delft University of Technology, Netherlands, E-mail:j.yang-3@tudelft.nl.*

1. https://www.internetlivestats.com/twitter-statistics/

Model training on positive-only datasets is typically achieved by leveraging expectation regularization techniques [7, 8]. In that context, the estimated proportion of relevant microposts in an unlabeled dataset containing a keyword is given as a *keyword-specific expectation*. This expectation is then used in the regularization term of the model's objective function to constrain the posterior distribution of the model predictions. By doing so, the model is trained with an expectation on its prediction for microposts that contain the keyword. The method for event detection proposed by Ritter et al.(2015), for instance, leverages expectation regularization; however, it suffers from two problems:

1) Due to the unpredictability of event occurrences and the constantly changing dynamics of users' posting frequency (Myers and Leskovec 2014), estimating the expectation associated with a keyword is a challenging task, even for domain experts;
2) The performance of the event detection model is constrained by the informativeness of the keyword used for model training. As of now, we lack a principled method for discovering new keywords and hence improve model performance.

Another major issue of previous work is the limitation of using keywords as an indicator of relevance [2, 10]. A keyword by itself is limited in its usefulness because of the lack of information it provides when characterizing event relevance in microposts. For example, for the predefined category of CyberAttack, the relevance of the keyword 'hack' in a micropost changes if another specific keyword like 'life' appears, compared to 'cyber' in the same micropost.

To address the above issues, we advocate a human-AI loop approach for discovering informative *rules* and estimating their expectations reliably. These rules are patterns in the microposts that encompass any features (not only

keywords) and can describe complex relationships between features using any logical operators. A rule is a simple statement consisting of a condition (also called antecedent) and a prediction. In our case, the antecedent can be any feature or combinations of features in a micropost, while the prediction is always an indication of event relevance. For example, a simple rule $(\text{'hack'} \cap \text{'Cyber'}) \implies \textit{event-category}\ (\text{expectation}) = 0.4$, states that if keywords 'hack' and 'cyber' are present in a micropost, then the expectation of this micropost being relevant to the event category of interest is 0.4.

This paper introduces an approach that iteratively leverages 1) crowd workers for estimating rule-specific expectations, and 2) the disagreement between the model and the crowd for discovering new informative rules. More specifically, at each iteration, we obtain a rule-specific expectation from the crowd by sampling a subset of the unlabeled microposts containing the rule and asking crowd workers to label these microposts. Then, we train the model using expectation regularization and select those rule-related microposts for which the model's prediction disagrees the most with the crowd's expectation; such microposts are then presented to the crowd to identify new rules that best explain the disagreement. By doing so, our approach identifies new rules which convey more relevant information with respect to existing ones, thus effectively boosting model performance. By exploiting the disagreement between the model and the crowd, our approach can make efficient use of the crowd, which is of critical importance in a human-in-the-loop context (Yan et al. 2011, Yang et al. 2018). An additional advantage of our approach is that by obtaining new rules that improve model performance over time, we can gain insight into how the model learns for specific event detection tasks. Such an advantage is particularly useful for event detection using complex models, e.g., deep neural networks, which are intrinsically hard to understand (Ribeiro et al. 2016; Doshi-Velez and Kim 2017).

We introduce a comprehensive set of strategies for effective rule discovery. First, as rules are inherently complex, we facilitate the process of rule discovery using an interactive interface where rules can be explored, and their utility be verified. Our interactive interface is useful for creating and verifying hypothesis about a relevant rule. Second, we automate the process of rule discovery by using decision trees. These decision trees are learned by leveraging the disagreement between the model's prediction and the crowd's expectation. Microposts with the highest disagreement are compared against microposts with the least disagreement to discover relevant rules automatically. An additional challenge during rule discovery is the unrealistic expectation from expert workers to have an exhaustive list of concepts and items for rule discovery. In this context, we explore the benefits of semantic enrichment through data augmentation [15, 16] and use it to augment our knowledge of the tweets' contents to give them a more meaningful feature representation. The data augmentation step is useful for expert workers, as they can also use these additional concepts obtained from data augmentation during rule discovery.

An additional challenge when involving crowd workers is that their contributions are not fully reliable (Vaughan 2018). In the crowdsourcing literature, this problem is usually tackled with probabilistic latent variable models (Dawid and Skene 1979; Whitehill et al. 2009; Zheng et al. 2017), which are used to perform truth inference by aggregating a redundant set of crowd contributions. Our human-AI loop approach improves the inference of keyword expectation by aggregating contributions not only from the crowd but also from the model. This, however, comes with its own challenge as the model's predictions are further dependent on the results of expectation inference, which is used for model training. To address this problem, we introduce a unified probabilistic model that seamlessly integrates expectation inference and model training, thereby allowing the former to benefit from the latter while resolving the inter-dependency between the two.

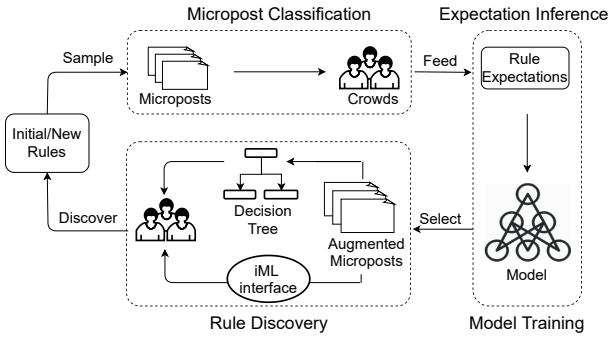


Fig. 1. An overview of our proposed human-AI loop approach. Starting from a (set of) new rule(s), it contains the following components: 1) Micropost Classification, which samples a subset of the unlabeled microposts containing the rule and asks crowd workers to label these microposts; 2) Expectation Inference & target model Training, which generates a rule-specific expectation and a micropost classification(target) model for event detection; 3) Rule Discovery, which applies the trained model and calculates the disagreement between target model prediction and the rule-specific expectation for discovering new rules. This is done through a hybrid human-machine workflow that engages human workers in discovering new rules via an interactive interface or verifying rules suggested by an automatic rule discovery component (here the decision tree).

In summary, we make the following key contributions:

1) A novel human-AI loop approach for micropost event detection that jointly discovers informative rules and estimates their expectation;
2) A unified probabilistic model that infers rule expectation and simultaneously performs machine learning model training;
3) A hybrid human-machine workflow that engages human workers in rule discovery through an interactive interface and leverages automatic methods for suggesting rules for human verification;
4) Semantic data exploration that augments the knowledge of expert workers by providing them with additional information related to tweets.

2 Human-AI Loop Workflow

Given a set of labeled and unlabeled microposts, our goal is to extract informative rules and estimate their expectation in order to train a machine learning model, referred to as target model in the paper. To achieve this goal, our proposed human-AI loop approach comprises two crowdsourcing

tasks, i.e., micropost classification followed by rule discovery, and a unified probabilistic model for both expectation inference and target model training. Figure 1 presents an overview of our approach. Next, we describe our approach from a process-centric perspective.

Following previous studies [2, 10, 3], we collect a set of unlabeled microposts $\mathcal{U}$ from a microblogging platform and post-filter, using an initial (set of) rule(s), those microposts that are potentially relevant to an event category. Then, we collect a set of event-related microposts (i.e., positively labeled microposts) $\mathcal{L}$, post-filtering them with a list of seed events. $\mathcal{U}$ and $\mathcal{L}$ are used together to train a discriminative model (e.g., a deep neural network) for classifying the relevance of microposts to an event. We denote the target model as $p_\theta(y|x)$, where θ is the model parameter to be learned and y is the label of an arbitrary micropost, represented by a bag-of-words vector x. Our approach iterates several times $t = \{1, 2, \ldots\}$ until the performance of the target model converges. Each iteration starts from the initial rule(s) or the new rule(s) discovered in the previous iteration. Given such a rule, denoted by $r^{(t)}$, the iteration starts by sampling microposts containing the rule from $\mathcal{U}$, followed by dynamically creating micropost classification tasks and publishing them on a crowdsourcing platform.

2.1 Micropost Classification

The micropost classification task requires crowd workers to label selected microposts into two classes: event-related and non event-related. In particular, workers are given instructions and examples to differentiate *event-instance* related microposts and *general event-category* related microposts. To understand the exact difference between those two classes, consider, for example, the following microposts, given in the context of *Cyber attack* events:

> *Credit firm Equifax says 143m Americans' social security numbers exposed in hack*

The micropost describes an instance of a cyber attack event that the target model should identify. It is, therefore, an *event-instance* related micropost and should be considered as a positive example. Contrast this with the following example:

> *Companies need to step their cyber security up*

This micropost, though related to cyber security in general, does not mention an instance of a cyber attack event, and is of no interest to us for event detection. This is an example of a general *event-category* related micropost and should be considered as a negative example.

For our task, each selected micropost is labeled by multiple crowd workers. The annotations are passed to our probabilistic model for expectation inference and target model training.

Expectation Inference & Model Training. Our probabilistic model takes crowd-contributed labels and the target model trained in the previous iteration as input. As output, it generates a rule-specific expectation, denoted as $e^{(t)}$, and an improved version of the micropost classification model, denoted as $p_{\theta^{(t)}}(y|x)$. The details of our probabilistic model are given in Section 3.3.

2.2 Rule Discovery

The rule discovery task aims at discovering a new rule (or, a set of rules) that is most informative for target model training with respect to existing rules. A useful rule consists of concepts and logical connections between them to indicate why (or, why not) a micropost belongs to an event category.

Formally, we aim to discover features that satisfy rules of the form:

$$features \implies \textit{Not event-category} \tag{1}$$

$$features \implies \textit{event-category} \tag{2}$$

In the above rules, features can be related to the presence or absence of relevant keywords, their combinations, the language of tweet, presence or absence of an entity, or an entity type; event-category is the event that we aim to discover. To illustrate the point further, the rule `('toll' ∩ 'death')` $\implies$ `Not PoliticianDeath` indicates that the tweet is likely to be irrelevant to *PoliticianDeath* event category. For example, the above rule applies to the tweet *'Death toll rises up to 100, PM will brief the conference'*; however, it does not indicate the death of a politician. As for any rule, the rule holds for the vast majority of the cases, but there can be exceptions.

Knowledge Augmentation. An additional important step to facilitate rule discovery is the knowledge augmentation of microposts so that expert workers can leverage the advantages of semantic enrichment when discovering rules. Knowledge augmentation of microposts is the process of enriching microposts with semantic annotations [21]. In the context of machine learning, models that have semantically meaningful representations are useful in helping humans make sense of the model behaviors [16]. Specifically, in classification problems, humans usually possess knowledge about the target class and can come up with hypotheses on the underlying concepts relevant to the problem.

We approach the idea of semantic annotations in the context of microposts. To accomplish this, we apply entity linking techniques to tag words and phrases with semantic annotations [22]. These inferred annotations then become part of the annotations of the microposts and can later be optionally used by the expert workers to form rules. This task is important as it is costly and unrealistic to ask humans to provide an exhaustive list of concepts. We explain this with an example further in *Rule Discovery Process*.

Rule Discovery Process. During rule discovery, we first apply the current target model $p_{\theta^{(t)}}(y|x)$ on the unlabeled microposts $\mathcal{U}$. For those that contain the rule $r^{(t)}$, we calculate the disagreement between the target model predictions and the rule-specific *expectation*, $e^{(t)}$:

$$Disagreement(x_i) = |p_{\theta^{(t)}}(y_i|x_i) - e^{(t)}|, \tag{3}$$

and select the ones with the highest disagreement for rule discovery. These selected microposts are supposed to contain information that can *explain* the disagreement between the target model prediction and rule-specific expectation, and can thus provide information that is most different from the existing set of rules for target model training.

For instance, our study shows that the expectation for the rule, `hack` $\implies$ `CyberAttack (expectation) =`

0.20, which means that only 20% of the initial set of microposts retrieved with the rule are event-related. A micropost selected with the highest disagreement (cf. Equation 3), whose likelihood of being event-related as predicted by the target model is 99.9%, is shown in the example below:

> *RT @xxx: Hong Kong securities brokers hit by cyber attacks, may face more: regulator #cyber #security #hacking https://t.co/rC1s9CB*

This micropost contains rules that can better indicate the relevance to a cyber security event, for e.g., ('cyber' ∩ 'hack') $\implies$ *CyberAttack* is more relevant than the initial rule ('hack') $\implies$ *CyberAttack*. Furthermore, using knowledge augmentation, we find that the tweet is referencing two entities from Wikipedia, 'Hacking (computer security)' and a location 'Hong Kong'. The presence of the augmented feature 'Hacking (computer security)' is important as it leads to tweets that talk about cyber hacking, and not just 'hacking' which could have another connotation. Besides, a rule formed using referenced entities like 'Hong Kong' could be useful to discover relevant tweets for a *CyberAttack* event associated with the location.

Note that when the rule-specific expectation, $e^{(t)}$ in Equation 3 is high, the selected microposts will be the ones that contain rules indicating the irrelevance of the microposts to an event category. Such rules are also useful for target model training as they help improve the model's ability to identify irrelevant microposts. For example, in the case of *PoliticianDeath* event, the rule ('innocent' ∩ 'bomb' ∩ 'explosion') $\implies$ *Not PoliticianDeath* usually indicated the event not related to the death of a politician but, an event where politician addressed a tragedy, which can be easily misclassified by an automatic classifier.

In this subsection, we explained how microposts showing disagreement with our target model (cf. Equation 3) help facilitate rule discovery. In the following subsection, we discuss two strategies to facilitate rule discovery by leveraging human input. The first strategy is through an interactive user interface, and the second is through the effective use of decision trees.

2.2.1 Expert Input via Interactive Interface

We use expert workers for rule discovery as the rules we consider are inherently complex and their utility needs to be verified.

One micropost can have multiple concepts, and a concept can be present in multiple microposts. In order to find relevant rules, it is important to be able to tease apart a concept (or, combination of concepts) and find microposts whose predictions do not align with the corresponding concepts. To achieve this, we propose an interactive visualization where microposts can be visualized with respect to the concepts they contain. We represent this relationship using a radial visualization where microposts are arranged inside the circle, and concepts are present on the circumference of the circle. Expert workers can use concepts as anchors to spread microposts based on the similarity between a micropost and selected anchors. As expert workers are supposed to be familiar with the concepts related to the chosen microposts, they can associate and contrast microposts' relations to the anchors (concepts).

Our interface design is explained in detail in Section 4.

2.2.2 Decision Trees

Decision trees are popular for their capability of learning interpretable rules (i.e., decision paths) [23, 24]. Along with rule discovery using our interactive interface, we also use decision trees to discover rules in an automated way. To facilitate rule discovery using decision trees, we leverage the difference between the tweets with the highest disagreement (cf. Equation 3) compared to those with the lowest disagreement. We build a decision tree using features from two classes - microposts with the lowest disagreement, and those with the highest disagreement with respect to rule expectation. As an example, during decision tree learning, the rule $micropost\_length < 100 \implies$ *Not CyberAttack* was generated as relevant. Before moving to the next step of expectation inference, automatically generated rules are verified by an expert worker.

3 Unified Probabilistic Model

This section introduces our probabilistic model that infers rule expectation and trains the target model simultaneously. We start by formalizing the problem and introducing our model, before describing the learning process.

Problem Formalization. We consider the problem at iteration t where the corresponding rule is $r^{(t)}$. In the current iteration, let $\mathcal{U}^{(t)} \subset \mathcal{U}$ denote the set of all microposts containing the rule and $\mathcal{M}^{(t)} = \{x_m\}_{m=1}^{M} \subset \mathcal{U}^{(t)}$ be the randomly selected subset of M microposts labeled by N crowd workers $\mathcal{C} = \{c_n\}_{n=1}^{N}$. The annotations form a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ where $\mathbf{A}_{mn}$ is the label for the micropost x_m contributed by crowd worker c_n. Our goal is to infer the rule-specific expectation $e^{(t)}$ and train the target model by learning the model parameter $\theta^{(t)}$. An additional parameter of our probabilistic model is the reliability of crowd workers, which is essential when involving crowdsourcing. Following Dawid and Skene [18, 20], we represent the annotation reliability of worker c_n by a latent confusion matrix $\boldsymbol{\pi}^{(n)}$, where the ab-th element $\boldsymbol{\pi}_{ab}^{(n)}$ denotes the probability of c_n labeling a micropost as class a given the true class b.

3.1 Expectation as Model Posterior

First, we introduce an expectation regularization technique for the weakly supervised learning of the target model $p_{\theta^{(t)}}(y|x)$. In this setting, the objective function of the target model is composed of two parts, corresponding to the labeled microposts $\mathcal{L}$ and the unlabeled ones $\mathcal{U}$.

The former part aims at maximizing the likelihood of the labeled microposts:

$$\mathcal{J}_1 = \sum_{i=1}^{L} \log p_\theta(y_i|x_i) + \log p_\sigma(\theta), \tag{4}$$

where we assume that θ is generated from a prior distribution (e.g., Laplacian or Gaussian) parameterized by σ.

To leverage unlabeled data for target model training, we make use of the expectations of existing rules, i.e., $\{(r^{(1)}, e^{(1)}), \ldots, (r^{(t-1)}, e^{(t-1)}), (r^{(t)}, e^{(t)})\}$ (Note that $e^{(t)}$ is inferred), as a regularization term to constrain model training.

To do so, we first give the target model's expectation for each rule $r^{(k)}$ $(1 \leq k \leq t)$ as follows:

$$\mathbb{E}_{x\sim\mathcal{U}^{(k)}}(y) = \frac{1}{|\mathcal{U}^{(k)}|}\sum_{x_i\in\mathcal{U}^{(k)}} p_\theta(y_i|x_i), \tag{5}$$

which denotes the empirical expectation of the target model's posterior predictions on the unlabeled microposts $\mathcal{U}^{(k)}$ containing rule $r^{(k)}$. Expectation regularization can then be formulated as the regularization of the distance between the Bernoulli distribution parameterized by the target model's expectation and the expectation of the existing rule:

$$\mathcal{J}_2 = -\lambda\sum_{k=1}^{t} D_{KL}[Ber(e^{(k)})\|Ber(\mathbb{E}_{x\sim\mathcal{U}^{(k)}}(y))], \tag{6}$$

where $D_{KL}[\cdot\|\cdot]$ denotes the KL-divergence between the Bernoulli distributions $Ber(e^{(k)})$ and $Ber(\mathbb{E}_{x\sim\mathcal{U}^{(k)}}(y))$, and λ controls the strength of expectation regularization.

Figure 2 depicts a graphical representation of our unified probabilistic model, which combines the target model for training (on the left) with the generative model for crowd-contributed labels (on the right) through a rule-specific expectation.

3.2 Expectation as Class Prior

To learn the rule-specific expectation $e^{(t)}$ and the crowd worker reliability $\boldsymbol{\pi}^{(n)}$ $(1 \leq n \leq N)$, we model the likelihood of the crowd-contributed labels $\mathbf{A}$ as a function of these parameters. In this context, we view the expectation as the class prior, thus performing expectation inference as the learning of the class prior. By doing so, we connect expectation inference with target model training.

Specifically, we model the likelihood of an arbitrary crowd-contributed label $\mathbf{A}_{mn}$ as a mixture of multinomials where the prior is the rule-specific expectation $e^{(t)}$:

$$p(\mathbf{A}_{mn}) = \sum_{b}^{K} e_b^{(t)}\boldsymbol{\pi}_{ab}^{(n)}, \tag{7}$$

where $e_b^{(t)}$ is the probability of the ground truth label being b given the rule-specific expectation as the class prior; K is the set of possible ground truth labels (binary in our context); and $a = \mathbf{A}_{mn}$ is the crowd-contributed label. Then, for an individual micropost x_m, the likelihood of crowd-contributed labels $\mathbf{A}_{m:}$ is given by:

$$p(\mathbf{A}_{m:}) = \sum_{b}^{K} e_b^{(t)}\prod_{n=1}^{N}\boldsymbol{\pi}_{ab}^{(n)}. \tag{8}$$

Therefore, the objective function for maximizing the likelihood of the entire annotation matrix $\mathbf{A}$ can be described as:

$$\mathcal{J}_3 = \sum_{m=1}^{M}\log p(\mathbf{A}_{m:}). \tag{9}$$

3.3 Unified Probabilistic Model

Integrating target model training with expectation inference, the overall objective function of our proposed model is given by:

$$\mathcal{J} = \mathcal{J}_1 + \mathcal{J}_2 + \mathcal{J}_3. \tag{10}$$

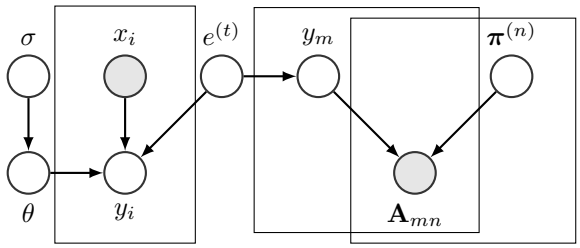


Fig. 2. Our proposed probabilistic model contains the target model (on the left) and the generative model for crowd-contributed labels (on the right), connected by rule-specific expectation.

Figure 2 depicts a graphical representation of our model, which combines the target model for training (on the left) with the generative model for crowd-contributed labels (on the right) through a rule-specific expectation.

4 Interactive Interface Design

In this section, we introduce our interactive interface. The objective of our design is to let the expert workers create rule hypotheses, and subsequently test their utility (i.e., informativeness for target model training).

The design of our interface is motivated by the field of interactive machine learning for error discovery, and it draws inspiration from previous work like AnchorViz and D&M [25, 26]. To facilitate rule discovery, we preselect frequently-occurring concepts in microposts. We also include the concepts obtained through micropost knowledge augmentation. Expert workers can look at all microposts related to a given concept (or, combinations of concepts) and discover relevant rules. When a relevant rule is selected, our interface filters out microposts that satisfy the rule. For example, an expert worker may discover a rule `('Wiki:Election' ∩ 'Wiki:Hacker(Computer security)')` $\Longrightarrow$ *`CyberAttack`*, as microposts filtered with this rule are likely to be relevant to *CyberAttack* event related to elections.

4.1 Workflow Design

The goal of the interface is to let expert workers create rules and validate their utility, which is supported by the following actions:

1) Let expert workers choose concepts for a given rule. First, the interface presents concepts to expert workers based on a chosen criterion. If needed, expert workers can also suggest a new concept that they deem as relevant.
2) The selected concepts are placed as anchors in the radial visualization (cf. Figure 3).
3) Microposts are spread around inside the radial visualization, based on the anchors. Anchors impact the positions of microposts in the radial visualization such that, microposts that are semantically closer to the anchors are situated closer to the anchors.
4) Microposts are colored according to the current target model prediction about the relevance or irrelevance of the micropost to an event category.

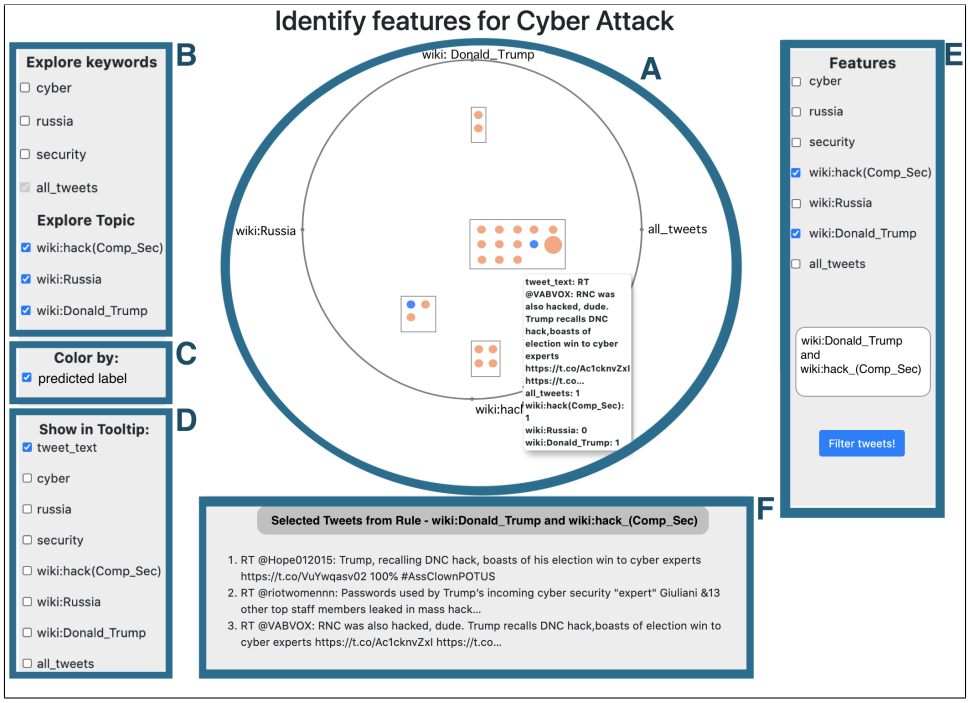


Fig. 3. Overview of our interface for rule discovery. The interface includes the radial visualization (A) and explore pane (B) that includes keyword based concepts and concepts obtained using knowledge augmentation. Based on the prediction of the model, the microposts can be colored. This is done using the color pane (C). Selecting multiple options from Tooltip pane (D) gives detailed information about each micropost features' when mouse is hovered over a micropost in radial visualization. Rules can be tested using the Feature selection option (E), filtered microposts are displayed in the results pane (F).

5) Let expert workers create rules and validate the utility of the rules with resulting microposts.

Next, we discuss the design details of the interface.

4.2 Interface Design

Figure 3 shows our interface, which is based on RadViz [27]. Microposts that have the most disagreement (cf. Equation 3) are arranged inside the radial visualization (A). Concepts associated with these microposts are arranged as anchors on the boundary of the radial visualization. When a concept is selected from the explore pane (B), the microposts inside the radial visualization rearrange themselves to reflect the update.

Topology and Layout Manipulation Once a concept(s) is selected, the expert worker can see the correlation between the concept(s) and the microposts as well as the relationship between several concepts with respect to the microposts. We map the relative similarity between the microposts and the anchors to the position of microposts in a non-orthogonal coordinate system on a circle; the center point of the circle to each anchor forms a set of axes on a 2D surface. Namely, an axis k is a vector with the length of the circle's radius r and an angle θ to the corresponding anchor.

$$\vec{V_k} = (r \times \cos\theta, r \times \sin\theta) \tag{11}$$

A micropost m along an axis forms a vector with an angle identical to that of the axis and $value_{k(m)}$, magnitude of the cosine similarity in the bag-of-words space between the micropost's concepts and the concept(s) represented by the anchor(s). The final position of a micropost m in the visualization is the sum of the vectors to each anchor.

$$(x_m, y_m) = \sum_k \frac{value_{k(m)}}{\sum_j value_j(m)} \times \vec{V_k} \tag{12}$$

In addition, to ensure that all items are inside the circle, a microposts' value on an axis is normalized by the sum of all its values on all the axes. We use the normalization introduce in RadViz [27] in that context.

This implies that the microposts that are closer to an anchor are more similar to the concept(s) represented by that anchor. The items that are affected by selecting or deselecting the anchor will change their positions accordingly. Items that do not share any similarity with the selected anchor will remain still. In this way, the expert workers can effectively create a topology in the concept space defined by the selected anchors. We considered multiple options but ultimately chose Radviz because of its support for an arbitrary number of axes and its flexibility in positioning axes while preserving the relative independence of the axes [28].

In the following subsections, we detail the features that can be used by an expert worker.

4.2.1 *Associating Model predictions with Concepts*

Our rule discovery process is based on the level of disagreement (cf. Equation 3) with the target model prediction. The color of the micropost in the visualization corresponds to the model prediction. By contrasting model predictions against anchors, expert workers can see which concept correlates with the presence, absence (or, indifference) of an event category. This helps expert workers to discover relevant rules for positive and negative model predictions.

4.2.2 Inspecting clusters of similar items

Microposts with similar concept representations are clustered into a rectangle. These clusters move along with the anchors that they are most similar to, just like individual microposts. When an expert worker hovers on a micropost inside this cluster, they can view all the concepts of the corresponding micropost that have been selected in the tooltip pane (cf. Figure 3 D).

4.2.3 Hypothesis creation and testing

A feature pane (Figure 3 E) can be used by an expert worker during rule discovery for hypothesis creation. Any rule hypothesis with the logical operators 'and', 'or', 'not' can be typed and tested using the feature pane. To test a rule hypothesis, an expert worker writes it in the feature pane. Microposts that satisfy this rule are updated in the selection pane (F). An expert worker validates the utility of the rule hypothesis by inspecting the relevance of resulting microposts to the event category. The combination of feature pane and selection pane supports rule hypothesis creation and testing.

5 Experiments and Results

This section introduces the experimental setup for evaluating our approach, followed by the results. Through our evaluation, we aim at answering the following questions:

- **Q1** How effectively does our proposed human-AI loop approach enhance the state-of-the-art machine learning models for event detection?
- **Q2** What is the effect of enriching microposts using knowledge augmentation?
- **Q3** What is the added advantage of introducing automated rule discovery methods as compared to rule discovery using an interactive interface alone?
- **Q4** How effective is our approach at obtaining new rules compared with an approach labeling microposts for target model training under the same cost?

5.1 Experimental Setup

Datasets. We perform our experiments with two predetermined event categories: cyber security (CyberAttack) and death of politicians (PoliticianDeath). We found that though there are a few publicly available datasets for this task, the available ones do not suit our requirements. For example, the publicly available Events-2012 Twitter dataset [29] contains generic event descriptions such as Politics, Sports, Culture, etc. Our work targets more specific event categories [15]. Following previous studies [2], we collected event-related microposts from Twitter using 11 and 8 seed events for *CyberAttack* and *PoliticianDeath*, respectively. Unlabeled microposts were collected by using the keyword 'hack' for *CyberAttack*, while for *PoliticianDeath*, we used a set of keywords related to 'politician' and 'death' (such as 'bureaucrat', 'dead' etc.). The dataset was collected for one year using Twitter public API. For each dataset, we randomly selected 500 tweets from the unlabeled subset and manually labeled them for evaluation. Table 1 shows key statistics from both datasets.

TABLE 1
Statistics of the datasets in our experiments.

Dataset	#Positive	#Unlabeled	#Test
CyberAttack	2,600	86,000	500
PoliticianDeath	900	7,000	500

Comparison Methods. We consider Logistic Regression (LR)[2] and Multilayer Perceptron (MLP)[3] as the target models[2]. These widespread models demonstrate the generality and effectiveness of our new model training technique.

For both LR and MLP, we evaluate our proposed approach for keyword discovery and expectation estimation by comparing against the weakly supervised learning method proposed by Ritter et al.(2015) which uses only one initial keyword with an expectation estimated by an individual worker and a LR model. Similarly, Chang et al.(2016) also used a neural model with one initial keyword with the same model training technique as Ritter et al. (2015). Wherever possible, we also compare to our previously proposed human-AI loop approach (Bhardwaj et al. 2020).

Parameter Settings. We empirically set optimal parameters based on a held-out validation set that contains 20% of the test data. These include the hyperparameters of the target model, those of our proposed probabilistic model, and the parameters used for training the target model. We explore MLP with 1, 2, and 3 hidden layers and apply a grid search in 32, 64, 128, 256, 512 for the dimension of the embeddings and that of the hidden layers. For the coefficient of expectation regularization, we follow Mann and McCallum (2007) and set it to $\lambda = 10\times$ #labeled examples. For target model training, we use the Adam [31] optimization algorithm for both models. We repeat the experiments 10 times and report the average results.

Evaluation. Following Ritter et al. (2015), Konovalov et al. (2017), we use accuracy and area under the precision-recall curve (AUC) metrics to measure the performance of our proposed approach. We note that due to the imbalance in the datasets (20% positive microposts in *CyberAttack* and 27% in *PoliticianDeath*), accuracy is dominated by negative examples; AUC—area under the precision-recall curve—in comparison, better characterizes the discriminative power of the model for imbalanced datasets. Higher values of accuracy and AUC indicate better performance.

Crowdsourcing. We have two categories of crowdsourcing tasks: rule discovery, and micropost classification. Rule discovery is performed by four expert workers who are our in-house participants as it requires domain expertise[3]. To avoid bias, the chosen expert workers are different for each experimental setting. Micropost classification is a binary classification task where the goal is to check if a micropost

2. Our experiments with large pretrained language models revealed that they are not suitable for our task and offer a lower increase in AUC as compared to LR and MLP (cf. Appendix A.2).

3. For a new task, one can consider finding experts in open crowdsourcing or social media platforms through expertise modeling and engagement, which are related research topics.

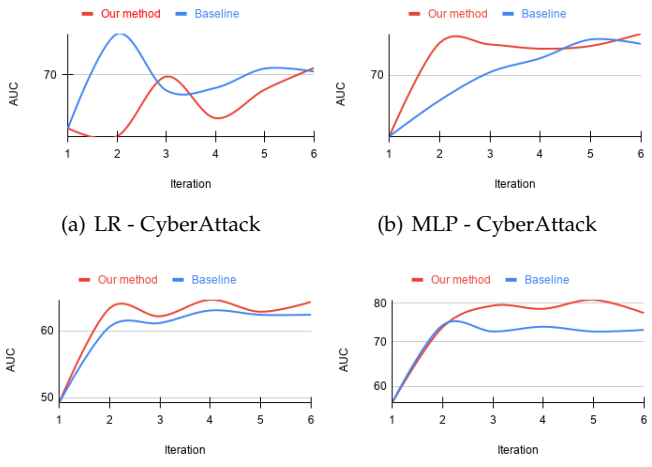
TABLE 2
Performance of the target models trained by our proposed human-AI loop approach on the experimental datasets at different iterations. Results are given in percentage.

Dataset	Method	Metric	Iteration					
			1	2	3	4	5	6
CyberAttack	LR	AUC	66.69	66.20	69.90	67.30	69.07	70.44
		Accuracy	71.04	72.39	71.04	71.38	72.39	71.04
	MLP	AUC	60.79	75.47	75.18	74.45	74.91	77.06
		Accuracy	70.37	74.07	74.07	74.07	74.07	75.08
PoliticianDeath	LR	AUC	49.37	63.85	62.46	65.33	63.23	64.95
		Accuracy	76.53	83.22	83.22	83.22	82.88	83.55
	MLP	AUC	56.81	73.71	79.4	78.52	81.01	77.37
		Accuracy	76.53	79.53	86.91	84.22	84.22	84.56

belongs to a relevant event category. It is performed by crowd workers[4]and does not require domain expertise.

For rule discovery, potentially relevant concepts are presented to our expert workers using our interactive interface. In-house participants then suggest rules, which are constructed using these concepts, their combinations with logical operators, and further any other rule that they find relevant to the task. We also use these concepts for decision tree-based automatic rule discovery. Our rule discovery process consisted of two parts: the first part (20 minutes) involved an introduction to basic machine learning knowledge such as classification, errors, precision and recall, description of the dataset, an overview of the interactive interface, and introduction of the event categories. The second part (10-20 minutes) was a practice round to get familiar with the interface, followed by an introduction to the *CyberAttack* and *PoliticianDeath* event categories, which are used for the actual task. The third part was the actual task where we asked the participants to use the interface to discover rules.

4. We have chosen Appen (https://appen.com/, formerly Figure-eight) as a crowdsourcing platform and have picked Level-3 workers which, correspond to the highest quality of crowdworkers.



(a) LR - CyberAttack (b) MLP - CyberAttack

(c) LR - PoliticianDeath (d) MLP - PoliticianDeath

Fig. 4. Comparison of AUC performance of our method against the baseline for *CyberAttack* (a) Logistic Regression (b) Multi Layer Perceptron. Comparison of AUC performance of our method against the baseline for *PoliticianDeath* (c) Logistic Regression (d) Multi Layer Perceptron.

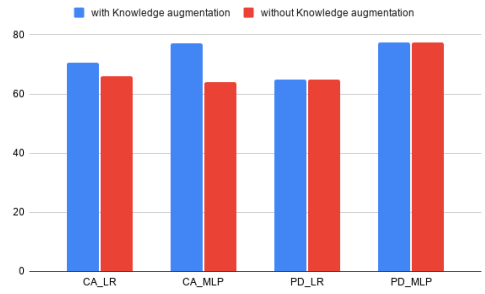
In terms of cost-effectiveness, our approach is motivated by the fact that crowdsourced data annotation can be expensive, and is thus designed with minimal crowd involvement. For each iteration, we selected 50 tweets for rule discovery and 50 tweets for micropost classification per rule. For a dataset with 80k tweets (e.g., *CyberAttack*), our approach only requires to manually inspect 600 tweets (for 6 rules), which is less than 1% of the entire dataset.

5.2 Q1: Comparison with SoTA

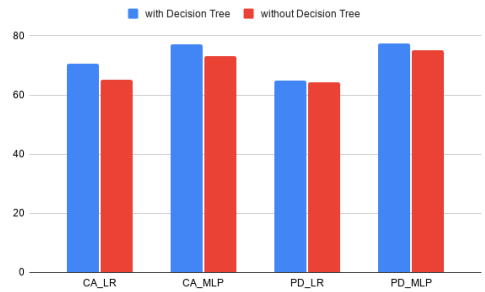
Table 2 reports the results of our approach on both the *CyberAttack* and *PoliticianDeath* event categories. Our approach improves LR and MLP by 9.17% and 8.6% in terms of accuracy, respectively, and by 19.79% and 31.48% in terms of AUC, respectively. Recall that AUC better characterizes the discriminative power of the model as accuracy is dominated by negative examples. Such significant improvements clearly demonstrate that our approach is effective at improving model performance.

Figure 4 shows a comparison of AUC performance against our previous keyword based approach [30]. We show the results for *CyberAttack* and *PoliticianDeath* event categories using the same LR and MLP models on the same dataset. Overall, we observe that results on *PoliticianDeath* dataset show more improvement using our approach. This is likely due to the fact that microposts that are relevant for *PoliticianDeath* are semantically more complex than those for *CyberAttack* as they enclose noun-verb relationship (e.g., 'the king of ... died ...') rather than a simple verb (e.g., '... hacked.') for the *CyberAttack* microposts. Our approach is useful here as rules (e.g., `'King' ∩ 'died' ⟹ PoliticianDeath`) are more effective than keywords at characterizing the death of a politician. We also note that in the majority of the cases, our approach reaches higher AUC scores in earlier iterations. This is because concepts related to a specific event instance were suggested by the crowd in earlier iterations when compared to our previous work [30]. For example, rules that contained concepts `'Donald_Trump'` and `'Election'` from Wikipedia were suggested as a relevant rule during the second iteration using our approach as compared to 'trump' and 'election' being discovered as keywords during the tenth and twelfth iterations using previous keyword discovery method. The unexpected low AUC performance in the initial iteration of LR model for *CyberAttack* shows that it is sometimes possible that a less useful rule can decrease the performance.

Despite a low AUC score in the earlier iterations for such a case, the performance improves as the target model is trained with novel information in further iterations.



(a) Effect of knowledge augmentation



(b) Effect of decision tree in rule-discovery

Fig. 5. Effective performance difference in terms of AUC with a) keyword augmentation b)Decision Trees. For brevity: CA → CyberAttack , PD → PoliticianDeath, LR → Logistic Regression, MLP → Multi-layer perceptron.

5.3 Q2: Effect of Knowledge Augmentation

In this subsection, we study the effects of our knowledge augmentation approaches. Figure 6 shows the quantitative difference in the AUC performance for both models from both event categories. The model with knowledge augmentation outperforms the model without knowledge augmentation by 13.62% in terms of AUC on *CyberAttack* but shows no difference on *PoliticianDeath*.

With knowledge augmentation, for the *CyberAttack* category, expert workers discovered a frequently occurring Wikipedia obtained concept, `'Hacker(Computer security)'` in the first iteration. This was clearly a better indicator of micropost's relevance to the event category than the initial keyword `'hack'`, which could be used in another context. Microposts filtered using these rules usually discussed the alleged election hack in the USA. Using these augmented concepts, expert workers were able to discover rules containing these concepts.

Interestingly for the *PoliticianDeath* event category, knowledge augmentation was not very useful. This is a more complex event category where merely the presence of `'politician'` and `'death'` does not indicate the death of a politician event. For example, a lot of microposts are about bomb attacks, explosions, protests where both `'politician'` and `'death'` keywords are present. Instead of knowledge augmentation, for this case, rules containing phrasal patterns were more helpful. For example, `'death to the dictator'` $\implies$ `Not PoliticianDeath` indicated protests in Iran, which was a useful rule and indicated the irrelevance to our target event category, *PoliticianDeath*.

5.4 Q3: Effect of Decision Trees

In this subsection, we study the effective advantage of using decision trees. Figure 5(b) shows the quantitative difference in performance obtained using decision trees. The model with decision trees consistently outperforms the model without decision trees across all datasets and models by an average of 4.42% in terms of AUC.

Decision trees generated about 20% of the rules. They are more useful to detect rules related to syntax that expert workers might miss. For example, in the case of *CyberAttack*, decision trees selected $micropost\_length < 100 \implies$ `Not CyberAttack`. When we further observed, this was quite true as the microposts with a length of fewer than 100 characters had relevant concepts like `'cyber'`, `'attack'`, `'security'`, but were mostly advertisements about an upcoming event. This rule was discovered only using decision trees and was quite useful to eliminate noisy microposts with relevant concepts.

For the case of *PoliticianDeath*, it was quite interesting that decision paths indicated a difference in the classes based on the singular/plural form of certain concepts. For example, `('ministers' ∪ 'politicians')` $\implies$ `Not PoliticianDeath`. When we looked further into this, we noticed that the plural form was usually associated with the absence of an event compared to the singular form. Though intuitive, this rule was discovered only using decision trees.

5.5 Q4: Cost effectiveness

In our previous work [30], we demonstrated the cost-effectiveness of using crowdsourcing for obtaining new keywords and consequently, their expectations, by comparing their performance with an approach using crowdsourcing to only label microposts for target model training at the same cost. Specifically, we conducted an additional crowdsourcing experiment where the same cost used for keyword discovery in our approach is used to label additional microposts for target model training. These newly labeled microposts are used with the microposts labeled in the micropost classification task (see Section *Micropost Classification*) and the expectation of the initial keyword to train the target model for comparison. The model trained in this way increases AUC by 0.87% for *CyberAttack*, and by 1.06% for *PoliticianDeath*; in comparison, our proposed approach [30] increases AUC by 33.42% for *PoliticianDeath* and by 15.23% for *CyberAttack* over the baseline presented by Ritter et al.[2]). These results proved that using crowdsourcing for keyword discovery is significantly more cost-effective than simply using crowdsourcing to get additional labels when training the target model.

With respect to the cost-effectiveness of the interactive interface in comparison to the previously presented keyword discovery approach, the upper cost bound in rule discovery is the cost if the expert worker chooses to go through each tweet one by one. The initial reported time for an expert worker to discover new rules was 24 minutes,

but once they got used to the task - it took an average of 15 minutes for each iteration (50 tweets)[5]. In comparison to this, during keyword discovery, a crowd worker spent 40 seconds on each tweet for each iteration. This cost in terms of time is at least twice better than our previous keyword discovery approach where a worker goes through each tweet one by one. With respect to the costs of decision tree-based rule discovery, automatic rule generation takes about 3 seconds as we are comparing only 50 selected microposts with highest disagreement against the ones with lowest disagreement. The top-generated rules are verified by an expert worker and the costs of verification is lower than the cost incurred for generating a new rule using interactive interface (which is itself low as seen above).

6 Related Work

6.1 Event Detection

The techniques for event extraction from microblogging platforms can be classified into three groups [1] based on domain specificity. The first group contains approaches for detecting unspecified events [32, 33], these are events of general interest but with no advance description. The second group contains approaches for detecting predetermined events, such as concerts, local festivals, earthquakes, and disease propagation [34, 35, 36]. The third group contains approaches for detecting specific events, which typically use IR methods to match a query [37, 38]. For example, 'Trump meets Obama'. Early works mainly focus on open domain event detection [36, 39, 40]. Our work falls into the category of domain-specific event detection [15], which has drawn increasing attention due to its relevance for various applications such as cyber security [2, 3] and public health [5, 6].

In terms of technique, our proposed detection method is related to other previously proposed weakly supervised learning methods [2, 10, 4]. These approaches come in contrast with fully-supervised learning methods, which are often limited by the size of the training data (e.g., a few hundred examples) [34, 41].

6.2 Human-in-the-Loop Approaches

Our work extends weakly supervised learning methods by involving humans in the loop (Vaughan 2018). Existing human-in-the-loop approaches mainly leverage crowds to label individual data instances (Yan et al. 2011; Yang et al. 2018) or to debug the training data (Krishnan et al. 2016; Yang et al. 2019) or components (Parikh and Zitnick 2011; Mottaghi et al. 2013; Nushi et al. 2017) of a machine learning system. Unlike these works, we leverage crowd workers to label sampled microposts in order to obtain rule-specific expectations, which can then be generalized to help classify microposts containing the same rule, thus amplifying the utility of the crowd. Our work is further connected to the topic of interpretability and transparency of machine learning models (Ribeiro et al. 2016; Lipton 2016; Doshi-Velez and Kim 2017), for which humans are increasingly involved, for instance for post-hoc evaluations of the model's interpretability. In contrast, our approach directly solicits informative rules from the crowd for model training, thereby providing human-understandable explanations for the improved model.

6.3 Neuro-symbolic Methods

Our method of integrating logic rules into machine learning is related to the current trend of AI research moving from machine learning to neuro-symbolic methods, and from data – to hybrid data – and knowledge-driven approaches. Those methods have shown to be more robust due to their capability in representing concepts and the causal relations among them, and have demonstrated their effectiveness for several tasks including health monitoring [5], document filtering [48], stock pricing [49]. Methodologically, there are mainly two approaches for integrating symbolic knowledge into neural networks. Xu et al. [50] introduce the semantic loss that augments the training objective of neural networks with soft-constraints specified with domain knowledge; Allamanis et al. [51] propose to learn continuous representations of symbolic knowledge for integration into neural networks. Those work, while providing methods for knowledge integration, does not discuss *what* knowledge to be integrated. Our approach of knowledge integration is similar to Xu et al. [50] while addressing specifically the discovery of most informative rules leveraging a hybrid human-AI approach.

6.4 Error Discovery

There are two primary strategies for searching items to label: machine-initiated and human-initiated approaches. The machine-initiated approaches use learning algorithms to suggest items for humans to label so that the model needs fewer training items to perform better, e.g., according to the uncertainty of the prediction by the model [52]. Such methods are, however, not suitable to identify errors produced with high confidence, namely unknown unknowns. Unlike machines that fully rely on knowledge explicitly encoded in predefined training data, humans excel at leveraging broad and tacit knowledge in justification. Human computation has, therefore, emerged as a major class of approaches to detecting unknown unknowns [53, 54]. Existing human computation methods mainly rely on humans to verify model predictions on a per-instance basis. In contrast, our approach involves humans to provide the reasons for model predictions that disagree with human expectation through a carefully designed rule discovery workflow, exploiting human intelligence in a more effective and efficient manner.

6.5 Visualization in ML

Interactive visualization in ML has been a key approach to facilitate model training [55]. Our design is based on a polar coordinated visualization method inspired by D&M [26] and Anchorviz [25]. D&M uses magnet metaphor to attract similar items using pre-defined dimensions of the data. Anchorviz is an interactive visualization interface that facilitates error discovery through data exploration. Our visualization is a part of our rule discovery step. It is inspired by two ideas of semantic exploration and decomposition of the dataset through anchors.

5. The interface loading and response time are excluded since they are negligible in our case, as the dataset is relatively small for those purposes.

7 Conclusion

In this paper, we presented a new human-AI loop approach for rule discovery and expectation estimation to better train event detection models. Our approach discovers informative rules and leverages the joint power of the crowd and the model in expectation inference. Our rule discovery method is a hybrid human-machine workflow that engages human workers through an interactive interface and leverages automatic methods for suggesting rules for human verification. We evaluated our approach on real-world datasets and showed that it significantly outperforms the state of the art, and is particularly useful for detecting events where categories are semantically complex, e.g., the death of a politician. In future work, we would like to explore how our approach scales with number of rules, size of datasets and kinds of event categories.

8 Acknowledgements

We thank the reviewers for their valuable feedback. This project has received funding from the Swiss National Science Foundation (grant #407540_167320 Tighten-it-All) and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement 683253/GraphInt).

References

[1] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, 2015.

[2] A. Ritter, E. Wright, W. Casey, and T. Mitchell, "Weakly supervised extraction of computer security events from twitter," in *WWW*, 2015.

[3] N. Chambers, B. Fry, and J. McMasters, "Detecting denial-of-service attacks from social media text: Applying nlp to computer security," in *NAACL*, 2018.

[4] A. Konovalov, B. Strauss, A. Ritter, and B. O'Connor, "Learning to extract events from knowledge base revisions," in *WWW*, 2017.

[5] M. Akbari, X. Hu, L. Nie, and T.-S. Chua, "From tweets to wellness: Wellness event detection from twitter streams." in *AAAI*, 2016.

[6] K. Lee, A. Qadir, S. A. Hasan, V. Datla, A. Prakash, J. Liu, and O. Farri, "Adverse drug event detection in tweets with semi-supervised convolutional neural networks," in *WWW*, 2017.

[7] G. S. Mann and A. McCallum, "Simple, robust, scalable semi-supervised learning via expectation regularization," in *ICML*, 2007.

[8] G. Druck, G. Mann, and A. McCallum, "Learning from labeled features using generalized expectation criteria," in *SIGIR*, 2008.

[9] S. A. Myers and J. Leskovec, "The bursty dynamics of the twitter information network," in *WWW*, 2014.

[10] C. Y. Chang, Z. Teng, and Y. Zhang, "Expectation-regulated neural model for event mention extraction," in *NAACL: Human Language Technologies*, 2016, pp. 400–410.

[11] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, "Active learning from crowds." in *ICML*, vol. 11, 2011.

[12] J. Yang, T. Drake, A. Damianou, and Y. Maarek, "Leveraging crowdsourcing data for deep active learning an application: Learning intents in alexa," in *WWW*, 2018.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *KDD*, 2016.

[14] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[15] A. Bhardwaj, A. Blarer, P. Cudre-Mauroux, V. Lenders, B. Motik, A. Tanner, and A. Tonon, "Event detection on microposts: a comparison of four approaches," *IEEE TKDE*, vol. PP, pp. 1–1, Oct. 2019.

[16] C. Jandot, P. Simard, M. Chickering, D. Grangier, and J. Suh, "Interactive semantic featuring for text classification," *arXiv preprint arXiv:1606.07545*, 2016.

[17] J. W. Vaughan, "Making better use of the crowd: How crowdsourcing can advance machine learning research," *JMLR*, vol. 18, no. 193, 2018.

[18] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, 1979.

[19] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS*, 2009.

[20] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?" *PVLDB*, 2017.

[21] I. Marinchev, "Semantic lifting of unstructured data based on nlp inference of annotations," in *Proceedings of the 13th ICCST*, 2012, pp. 58–63.

[22] F. Piccinno and P. Ferragina, "From tagme to wat: a new entity annotator," in *Proceedings of the first international workshop on Entity recognition & disambiguation*, 2014, pp. 55–62.

[23] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *KDD*, 2016, pp. 1675–1684.

[24] B. Nushi, E. Kamar, and E. Horvitz, "Towards accountable ai: Hybrid human-machine analyses for characterizing system failure," in *HCOMP*, vol. 6, no. 1, 2018.

[25] J. Suh, S. Ghorashi, G. Ramos, N.-C. Chen, S. Drucker, J. Verwey, and P. Simard, "Anchorviz: Facilitating semantic data exploration and concept discovery for interactive machine learning," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 10, no. 1, pp. 1–38, 2019.

[26] J. Soo Yi, R. Melton, J. Stasko, and J. A. Jacko, "Dust & magnet: multivariate information visualization using a magnet metaphor," *Information visualization*, vol. 4, no. 4, pp. 239–256, 2005.

[27] P. E. Hoffman, *Table visualizations: a formal model and its applications*. University of Massachusetts Lowell, 2000.

[28] J. Sharko, G. Grinstein, and K. A. Marx, "Vectorized radviz and its application to multiple cluster datasets," *IEEE transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1444–1427, 2008.

[29] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on twitter," in *CIKM*. ACM, 2013, pp. 409–418.

[30] A. Bhardwaj, J. Yang, and P. Cudré-Mauroux, "A human-ai loop approach for joint keyword discovery and expectation estimation in micropost event detection." in *AAAI*, 2020, pp. 2451–2458.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[32] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," 2011.
[33] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu, "Towards effective event detection, tracking and summarization on microblog data," in *International conference on web-age information management*. Springer, 2011, pp. 652–663.
[34] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *WWW*, 2010.
[35] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection," in *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*, 2010, pp. 1–10.
[36] E. Benson, A. Haghighi, and R. Barzilay, "Event discovery in social media feeds." ACL, 2011.
[37] M. Rospocher, M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger, and T. Bogaard, "Building event-centric knowledge graphs from news," *Journal of Web Semantics*, vol. 37, pp. 132–151, 2016.
[38] A. Tonon, P. Cudré-Mauroux, A. Blarer, V. Lenders, and B. Motik, "Armatweet: detecting events by semantic tweet analysis," in *ESWC*. Springer, 2017, pp. 138–153.
[39] A. Ritter, O. Etzioni, S. Clark *et al.*, "Open domain event extraction from twitter," in *KDD*, 2012.
[40] F. Chierichetti, J. M. Kleinberg, R. Kumar, M. Mahdian, and S. Pandey, "Event detection via communication pattern analysis." in *ICWSM*, 2014.
[41] M. Sadri, S. Mehrotra, and Y. Yu, "Online adaptive topic focused tweet acquisition," in *CIKM*, 2016.
[42] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "Activeclean: interactive data cleaning for statistical modeling," *PVLDB*, vol. 9, no. 12, pp. 948–959, 2016.
[43] J. Yang, A. Smirnova, D. Yang, G. Demartini, Y. Lu, and P. Cudré-Mauroux, "Scalpel-cd: leveraging crowdsourcing and deep probabilistic modeling for debugging noisy training data," in *WWW*, 2019.
[44] D. Parikh and C. Zitnick, "Human-debugging of machines," *Second NIPS Workshop on Computational Social Science and the Wisdom of Crowds*, vol. 2, no. 7, p. 3, 2011.
[45] R. Mottaghi, S. Fidler, J. Yao, R. Urtasun, and D. Parikh, "Analyzing semantic segmentation using hybrid human-machine crfs," in *CVPR*, 2013.
[46] B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann, "On human intellect and machine failures: Troubleshooting integrative machine learning systems." in *AAAI*, 2017.
[47] Z. C. Lipton, "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490*, 2016.
[48] J. Proskurnia, R. Mavlyutov, C. Castillo, K. Aberer, and P. Cudré-Mauroux, "Efficient document filtering using vector space topic expansion and pattern-mining: The case of event detection in microposts," in *CIKM*, 2017.
[49] Y. Yoon, T. Guimaraes, and G. Swales, "Integrating artificial neural networks with rule-based expert systems," *Decision Support Systems*, vol. 11, no. 5, pp. 497–507, 1994.
[50] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Broeck, "A semantic loss function for deep learning with symbolic knowledge," in *ICML*. PMLR, 2018, pp. 5502–5511.
[51] M. Allamanis, P. Chanthirasegaran, P. Kohli, and C. Sutton, "Learning continuous semantic representations of symbolic expressions," in *ICML*. PMLR, 2017, pp. 80–88.
[52] B. Settles, "Active learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 6, no. 1, pp. 1–114, 2012.
[53] J. Attenberg, P. Ipeirotis, and F. Provost, "Beat the machine: Challenging humans to find a predictive model's "unknown unknowns"," *(JDIQ)*, vol. 6, no. 1, pp. 1–17, 2015.
[54] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz, "Identifying unknown unknowns in the open world: representations and policies for guided exploration," in *AAAI*, 2017, pp. 2124–2132.
[55] J. A. Fails and D. R. Olsen Jr, "Interactive machine learning," in *IUI*, 2003, pp. 39–45.
[56] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," *arXiv preprint arXiv:1605.07891*, 2016.
[57] S. Kuzi, A. Shtok, and O. Kurland, "Query expansion using word embeddings," in *CIKM*, 2016.

Akansha Bhardwaj is a PhD candidate at eXascale Infolab, University of Fribourg in Switzerland supported by the Swiss National Foundation (SNF). Her current research focuses on detection of important events in different kinds of media, and she works with Prof. Dr. Philippe Cudre-Mauroux. Akansha has a M.Sc in Computer Science (*Intelligent Systems*) from Technical University, Kaiserslautern in Germany.

Jie Yang is Assistant Professor in the Web Information Systems group at Delft University of Technology and Co-Director of the Delft Design@Scale AI Lab. Before, he worked as a scientist at Amazon and a senior researcher at the eXascale Infolab, University of Fribourg. His research interests include Human-Centered AI, Crowd Computing, and Natural Language Processing. His current work focuses on developing human-in-the-loop methods and tools for the design and development of trustworthy AI systems.

Philippe Cudré-Mauroux is a Full Professor and the director of the eXascale Infolab at the University of Fribourg in Switzerland. Previously, he was a postdoctoral associate working in the Database Systems group at MIT. He received his PhD from the Swiss Federal Institute of Technology EPFL, where he won both the Doctorate Award and the EPFL Press Mention in 2007. Before joining the University of Fribourg, he worked on distributed information and media management for HP, IBM Watson Research (NY), and Microsoft Research.