A Study on Interaction in Human-In-The-Loop Machine Learning for Text Analytics

Yiwei Yang
University of Michigan, Ann Arbor
yanyiwei@umich.edu

Eser Kandogan
IBM Research, Almaden
eser@us.ibm.com

Yunyao Li
IBM Research, Almaden
yunyaoli@us.ibm.com

Prithviraj Sen
IBM Research, Almaden
senp@us.ibm.com

Walter S. Lasecki
University of Michigan, Ann Arbor
wlasecki@umich.edu

ABSTRACT

Machine learning (ML) models are often considered "blackboxes" as their internal representations fail to align with human understanding. While recent work attempted to expose the inner workings of ML models they do not allow users to interact directly with the model. This is especially problematic in domains where labeled data is limited as such the generalizability of ML models becomes questionable. We argue that the fundamental problem of generalizibility could be addressed by making ML models explainable in abstractions and expressions that make sense to users and by allowing them to interact with the model to assess, select, and build on. By involving humans in the process this way, we argue that the co-created models will be more generalizable as they extrapolate what ML learns from few data when expressed in higher level abstractions that humans can verify, update, and expand based on their domain expertise. In this paper, we introduce RulesLearner that expresses ML model as rules on top of semantic linguistic structures in disjunctive normal form. RulesLearner allows users to interact with the patterns learned by the ML model, e.g. add and remove predicates, examine precision and recall, and construct a trusted set of rules. We conducted a preliminary user study which suggests that (1) rules learned by ML are explainable and (2) co-created model is more generalizable (3) providing rules to experts improves overall productivity, with fewer people involved, with less expertise. Our findings link explainability and interactivity to generalizability, as such suggest that hybrid intelligence (human-AI) methods offer great potential.

CCS CONCEPTS

• **Human-centered computing** → **User studies**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Explainable AI; Interactive Machine Learning; Hybrid Intelligence Systems; Human-in-the-Loop ML

ACM Reference Format:
Yiwei Yang, Eser Kandogan, Yunyao Li, Prithviraj Sen, and Walter S. Lasecki. 2019. A Study on Interaction in Human-In-The-Loop Machine Learning for Text Analytics. In *Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019*. ACM, New York, NY, USA, 7 pages.

IUI Workshops'19, March 20, 2019, Los Angeles, USA.

1 INTRODUCTION

As machine learning models advance and become widely used in a variety of domains, they have also become more complex and difficult to understand. They are often viewed as "blackboxes", because their inner representation often have large number of parameters (tens of millions) and these parameters don't necessarily align with human understanding. As such, humans cannot hope to even begin to interpret the interplay and how it achieves its final prediction.

While model performance is important, explainability often matters more, especially in high-stake domains, such as predicting in-hospital mortality, where users would want to understand why a decision is made [8, 9]. Explainability is further more critical in domains where labeled data is limited due to ownership, privacy, or regulations, where ML models tend to overfit to data at hand [1]. The premise of explainability is to make the models understandable such that users can diagnosize and refine the model [5].

Recent work has shown that by explaining each prediction to the user, user would be able to provide necessary correction back to the system [6]. However, such human-model interaction is limited as people cannot directly modify the inner representation of the model. The key insight in this paper is that, by making ML models explainable in abstractions and expressions that make sense to users, we enable them to interact with the model directly to assess, select, and build on the learned rules. By involving humans in the process this way we argue that the co-created model will be more generalizable as it extrapolates what ML learns from few data when expressed in higher level abstractions, so that humans can verify, update, and expand based on their domain expertise.

In this work, we propose a hybrid human-machine intelligence approach which first learns a white box model in the form of rules that people can understand, and then employs domain experts to enhance the model by selecting or perhaps even updating the rules that can generalize beyond the data trained. We built a user interface in RulesLearner, to allow users to quickly explore and understand the rules for text analytics, specified as semantic linguistic structures in disjunctive normal form. Through RulesLearner, users can rank and filter rules by performance measures (precision, recall, F1), read examples, decompose rules into its predicates, and update rules by adding or dropping predicates. Since the users are experts, they have the necessary domain knowledge to determine whether

or not each learned rule makes sense. Hence, by selecting or updating rules, they can create a trusted model that generalizes over never-before-seen data.

We evaluated RulesLearner with 6 participants to examine the interaction between people and an explainable model. Our results show that, within 30 minutes, participants are able to co-create a model that outperforms a black box model (LSTM) on F1 by a range of 35.58% to 43.72%. When compared to experts who write rules from scratch, we found out that participants with the RulesLearner are able to achieve a better recall (88.64%) and F1 (58.75%) on average. Our results link explainability and interactivity to generalizability, as such suggest that hybrid intelligence (human-AI) can potentially do better than either human or machine alone.

The contributions of this paper are the following:

- a hybrid approach of training a white box model and allowing human to interact, review, modify it
- RulesLearner, the system that facilitates the exploration and evaluation of rules
- evaluation of the approach through a user study with text analytics software engineers

2 FORMATIVE STUDY

To inform our approach, we interviewed (2) expert text analytics researchers and (4) developers, in one-to-one and group sessions, on their practices, tools, and challenges when developing rules to classify sentences into categories for text extraction.

Essentially, current practice amounts to an iterative approach, in which they would study the domain of interest, identify concepts and relations, manually examine documents for variety of ways they are expressed, sketch semantic patterns matching these concepts and relations, transform patterns to a text extraction language (e.g. regular expressions to semantic role labels), execute them over sample labeled data, examine results and output, and iterate over, until they can convince themselves that this could work in a robust-enough way, running regression tests throughout the process. While they measure performance quantitatively on test datasets, they also examine lots of unlabeled documents just to look for elements that they should have picked up. The whole process often takes several weeks for a team of software engineers.

Improving recall (the fraction of true positives over all true instances) is identified to be by far the most difficult part of the whole process. One researcher stated, "*Identifying and fixing precision errors tends to be relatively easy. Thinking of additional ways that something can be expressed is a lot more difficult because you don't necessarily know what to look for. You are extrapolating from labeled data and labeled data is hard to come by...*" The scale of the data poses serious challenges for experts, "*even if you have all labeled data you still need help, cause you can't just read through all.*" As for experiences with machine learning so far, the researchers commented that it is not easy to obtain training data (corporate contracts) in their domain, so machine learning models typically overfit and do very poorly on real-world datasets.

In summary, we identified two challenges for human to create rule-based models: (1) Achieving high recall is very difficult, because humans do not know what pattern to look for among the labeled datasets. (2) Developing rules is time consuming, because humans need to manually look through massive datasets. These challenges motivated us to introduce a hybrid approach where AI can go through large datasets to create rules that humans can validate, update, and build on.

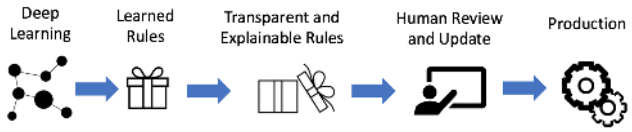

Figure 1: Overview of the Approach

3 APPROACH

Fig. 1 shows an overview of our approach where we begin by learning rules from labeled data using deep learning followed by using our system to explain said rules to domain experts so they can understand, ratify and possibly, modify these into potentially better rules. In this section, we will first briefly explain the Semantic Linguistic Structure used to learn rules and how we learned rules. We will then describe the features of our system in detail, and then demonstrate the usages in a hypothetical scenario.

Semantic Linguistic Structure (SLS) We learn rules on top of SLS that refer to the shallow semantic representations of sentences generated automatically with natural language processing techniques such as semantic role labeling [7] and syntactic parsing [4]. Such SLS are interpretable by human experts and captures "who is doing what to whom, when, where and how" described in a sentence as depicted by the following example (simplified for readability).

$$\underbrace{\text{John}}_{\texttt{agent}}\ \underbrace{\text{bought}}_{\texttt{action}}\ \underbrace{\text{daisy}}_{\texttt{theme}}\ \underbrace{\text{for Mary}}_{\texttt{beneficiary}}\ \underbrace{\text{yesterday}}_{\texttt{context:temporal}}\ .$$

One may use such SLS as predicates to build rules. For instance, the rule $\texttt{tense}(\text{action}) = \texttt{past} \wedge \text{action} \in \texttt{BuyingVerbsDictionary}$ holds true for sentence examples that depict the *buy* action in past tense (note that multiple actions may be present within the same sentence).

Rule Learning Learning rules has been a long-standing research task for which recent work has resorted to deep learning [3, 10]. To avoid overfitting on the training data, these works constrain the power of deep learning by introducing inductive bias presented in the form of SLS-based predicates to learn models in the form of interpretable rules. While reduced, the risk of overfitting however is not completely eliminated especially when labeled training data is scarce, to which end we propose to expose the learned rules to domain experts who can modify them into rules that better generalize to unseen data. Since we are more interested in studying the interaction between humans and explainable models, we refer the interested reader to the above references for the precise details of the learning algorithm which are out of scope of this paper.

3.1 User Interface

Our system allows people to interact with machine-learned rules and facilitate cooperative model development. There are two primary challenges: (1) present users with a quick overview of learned rules; enable them to organize, order, and navigate rules effectively,

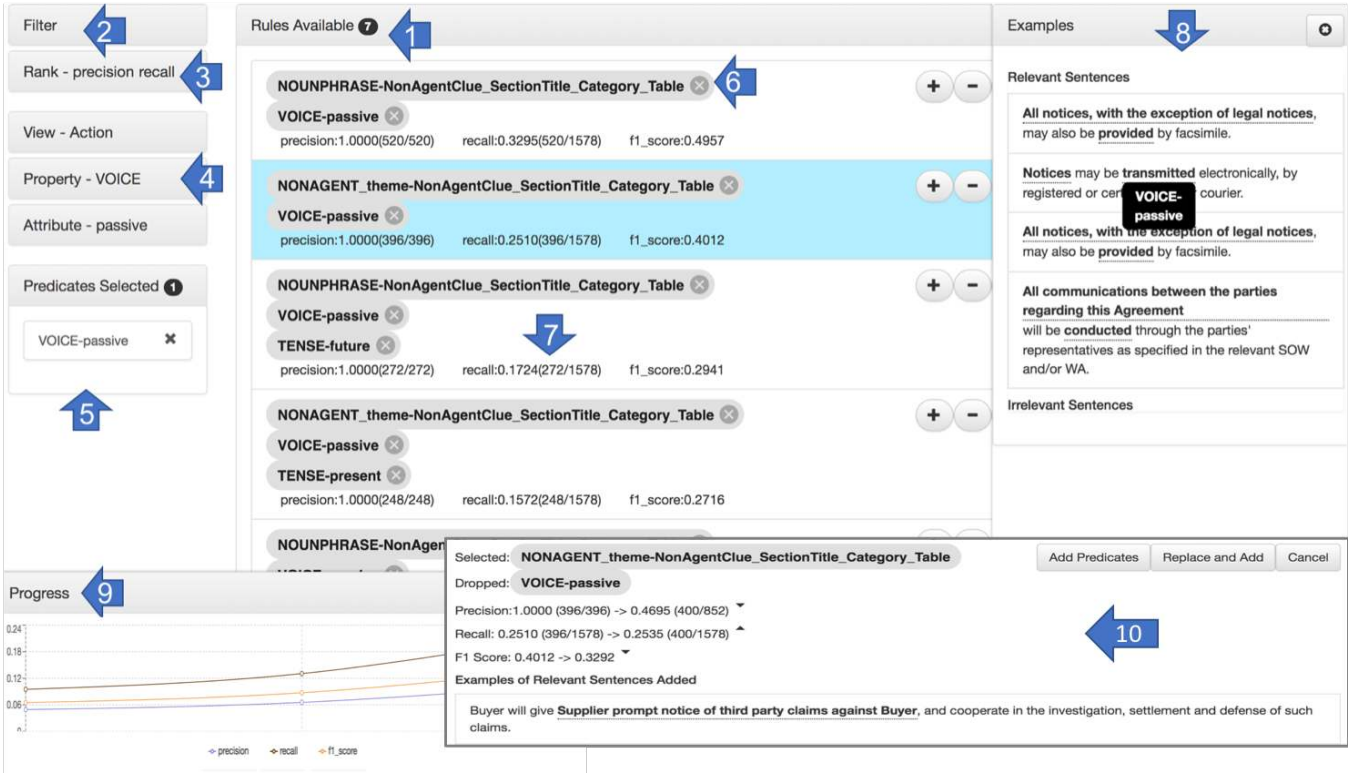

Figure 2: UI allows users (1) to get an overview of rules (2) filter by precision, recall, and F1, (3) rank, (4,5) filter by predicates, (6) remove rules by predicate, (7) examine metrics and (8) examples for each rule, (9) monitor overall progress as users add and remove rules to their collection, and (10) provide a 'playground' allowing users to examine and modify rules.

(2) help understand each rule's semantics and quality through examples and statistics; deepen understanding by providing a 'playground' to verify and modify rules while examining impact (Fig. 2).

3.1.1 Ranking. Ranking allows users to quickly see the rules with high performance on training data and process recommended rules in order. Each rule has its precision, recall, and f1 score, which are calculated by running the individual rule on the training data. To rank rules, users can choose a metric to sort by in the Ranking panel, which would order the rules in decreasing order according to the selected measure. Users can also rank by a secondary measure by selecting an additional rank order. This is useful especially when the list of rules is enormous, causing many rules to tie on the first selected measure.

3.1.2 Filtering. Filtering allows people to quickly narrow down to a small subset of similar rules, and focus on evaluating those rules without being overwhelmed or distracted. Users can filter rules by setting a a minimum threshold on several performance measures using the scrollbars, one for each measure. Users can also filter rules by predicates. As each rule is composed of one or multiple predicates, they can filter rules by clicking on the predicate within the rule. This is useful when the users reckon a rule potentially generalizable, and would like to see similar rules that share one or more common predicates.

3.1.3 Adding / Removing. The end goal of the system is to create a collection of trusted rules. After evaluating a rule, users can click the '+' or '-' button to add the rule to the 'approved' or 'disapproved' collection of rules, respectively. Each time a rule gets approved, the performance measures of the 'approved' collection are recalculated. The overall collection classifies an instance 'true' (retrieve from now on) for a category if any rule in the collection retrieves it, otherwise it classifies it 'false'. By comparing the predictions against the ground truth, we obtain true positives, false positives, and false negatives, which are used to calculate precision, recall, and F1. This helps users to keep track of the remaining rules and their overall progress.

3.1.4 Batch Removing. As users evaluate rules, they may discover rules that overfit by realizing one or more predicates do not make sense for the task. Our UI provides the ability to remove rules in batches which have such predicates. To do so, they can click on the 'X' button next to a predicate, and then click on the trash button. This feature can help users quickly prune out the overfitting rules en masse, leaving more potentially useful rules.

3.1.5 Examples. To help users assess the rules, our UI provides matching example sentences. As each rule is trained on a large number of sentences, enumerating all of them would be overwhelming.

Instead, we provide a random sample of up to 4 relevant (true positives) and 4 irrelevant (false positives) examples. Each example sentence highlights the matching predicates in the rule, and show the name of the predicate, when hovered (Fig. 2).

3.1.6 Look Ahead. Our UI provides a "look ahead" feature that allows users to see the effect of adding a rule to the existing collection of trusted rules. To "look ahead", users can simply hover over the Effect button next to the rule. Then, they will see the change in performance measures if such a rule gets added.

Although we show the performance measures underneath each rule, these numbers may be misleading as the sentences retrieved by the rule may be already covered by other rules in the collection. So the 'Look Ahead' function is a second glimpse people can take to quickly see the quality of the rule, and determine if they would like to take a close look by reading its associated examples, or examine it in 'Playground' mode.

3.1.7 Playground. Our UI provides a Playground mode to allow users inspect and modify rules by adding or dropping predicates, and examine its effects. To activate the Playground mode, users need to click the Inspect button on the rule.

In the Playground, a rule is decomposed into individual predicates and are shown in two sections: Selected and Dropped. Initially, all predicates are in the Selected section. When a predicate in the Selected section is clicked, it is moved over to the Dropped section, and the performance measures are updated accordingly, shown under each section. Conversely, if people click on a dropped predicate, then it will be added back to the Selected section, again updating the measures.

Users can also examine examples as they play with the rules. The Example sections show the "delta" examples, that is, if a predicate gets dropped, then the rule becomes more general, so it will retrieve more sentences, compared to the original rule. Conversely, if a predicate is added, it will retrieve fewer sentences, and we will show examples of the difference. This is beneficial because it allows users to see the effect of individual predicates. For example, if they believe a predicate is not necessary, yet they cannot quite decide solely based on performance measures, they can drop the predicate and verify whether or not more true positives or false positives are being retrieved, thus gaining a deeper understanding.

In the Playground, users can also add new predicates that are not part of the original rule. This is especially useful if experts already have a sense of what predicates are potentially good. To try them out they can click on the 'Add Predicate' button, and then choose from the list of all predicates to construct a new rule. Performance measures and examples are updated accordingly, helping users decide whether or not to keep the modification.

3.2 Scenario

Jan, an expert NLP engineer, wants to build a model for corporate contracts. Specifically, she wants to analyze and extract sentences related to 'Communication', i.e. exchange of information among parties in contracts.

Jan starts off with 188 machine-generated rules. She first *ranks* rules by precision and recall, and then *filters* by setting minimal thresholds, quickly cutting down the number of potentially interesting rules to 11. Noting that the top rule has reasonably high precision and recall, she decides to examine it further. She reckons that the matching *examples* belong to 'Communication', yet she is unsure about one particular predicate. She opens the *Playground*, and drops this predicate to examine its impact. She figures that, while precision remains roughly constant, recall increases. She also notices that the additional sentences retrieved by the new rule are valid matches. She then *adds* the updated rule to her collection. Noting that the dropped predicate does not make sense for 'Communication', she *batch removes* all rules containing that predicate. From her previous experience, Jan knows that a particular predicate could be useful, so *filters* rules by the predicate to examine them. She hovers over Effect on the top rule to *look ahead* the performance measures, and reckons that it may potentially generalize well but wants to examine further. She opens up the *Playground* and after adding and removing predicates, she feels confident to *add* the new rule based on all the *examples*. She then iterates through additional rules until she is satisfied with precision and recall of the model.

4 STUDY

4.1 Hypotheses

We hypothesize that our approach: (H1) creates an *explainable* model, (H2) allows users to co-create a more *generalizable* model, and (H3) increases *productivity* of the rule development process. To examine these hypotheses, we designed three experimental conditions, as explained below.

Baseline $B1$ is the machine-learning-only approach, trained on a dataset of sentences from corporate contracts, using Long Short-term Memory Model (LSTM), one of the most competitive machine learning models for the task.

Baseline $B2$ is the human-only approach, where (2) expert users develop a rule-based model from scratch, given limited tooling support. $B2_a$ has the same level of expertise as the participants in the treatment condition, and $B2_b$, the super expert, is much more experienced, with above 10 years of rule development experience in industry.

Treatment group $T1$ is the human-machine co-operative approach, where (4) expert users co-create a rule-based model, by examining machine generated rules, adding and removing predicates, as necessary, to build a trusted collection of rules.

We hypothesize that the human-machine co-operative approach would allow our experts ($T1$) to develop rules better than the expert ($B2_a$) with the same level of expertise, while the super expert ($B2_b$) is considered as an upper-bound, a goal. As such we measured how close, if not better, would the experts achieve with our system.

Typically, engineers develop AQL [2] rules by first examining examples, writing rules based on the patterns they found, compiling and running the rules to test on different validation sets. While our tool would clearly save time, we are not interested in measuring the low-level development time saved. Hence, we built a tool for human-only condition ($B2$) to easily select predicates to construct rules, see examples (relevant and irrelevant), examine dictionaries, and measure performance (precision, recall, and F1 score). Essentially it

is the same UI as in the treatment condition, without the machine-generated rules. As such, it allows us to focus on their ability to find and refine patterns without the assistance of machine intelligence.

Comparing $T1$ to $B1$ allows us to measure the ability of our approach to generate an explainable and generalizable model. Comparing $T1$ to $B2$ allows us to study human-machine cooperation, and evaluate the effect of our system.

4.2 Participants

6 engineers specialized in developing rule-based models for text extraction were recruited. While we consider them all as experts, their level of expertise varied. One expert is designated as a super expert, as she has over 10 years of experience.

4.3 Procedure

In both $B2$ and $T1$ conditions subjects were briefed about the study goals and were asked for permission to record the experiment, who all agreed. Each section was conducted individually and lasted anywhere from 1 to 1.5 hours. At the beginning of each session we gave the subjects documentation on rules and the relevant system. The documentation clarified the difference between the ML rules and the AQL rules, which subjects are familiar with. We also gave them hands-on demo, and allowed them to ask any questions. Each session included a pre-study where subjects were given ten rules and were asked to describe what the rule does in plain English. We then conducted the main experiment where subjects were asked to build a rule-based model, until they felt comfortable with the model performance, limited max. to an hour. During the experiment, we recorded the computer screen and audio and asked the subjects to speak aloud. Finally, we conducted a short interview on their experience.

4.4 Dataset

We conduct experiments on a dataset consisting of proprietary, legal contracts. The training set contains 28174 sentences extracted from 149 IBM procurement contracts while the test set has 1259 sentences extracted from 11 non-IBM procurement contracts. From these sentences, we extract 6 distinct SLS such as verbs, noun-phrases and agents (do-er of the verb). We use predicates of two flavours: predicates that test properties such as tense and predicates that check presence of extracted SLS in dictionaries. With 6 such property-based predicates and by looking up the different SLS in 33 distinct dictionaries we generated a total of 183 predicates.

4.5 Task

The task is to build a model for binary text classification. Specifically, the experts were asked to develop a rule-based model that determines whether or not the sentences from corporate contracts belong to the 'Communication' category, defined in the Scenario section.

4.6 Performance Measures

To assess success of our approach, we measured the precision, recall, and F1 score of the models created in B1, B2, and C1. To do this, we ran our models on the held-out test data set and compared the predictions of the models against the ground truth labels, which gave us the number of true positives(tp), false positives(fp), and false negatives(fn).

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

We get F1 by taking the harmonic mean of precision and recall.

5 RESULTS

5.1 A generalizable model

To measure generalizability we compared F1 scores on held-out test sets in our human-machine cooperative treatment group, $T1$, and machine-learning only $B1$ (LSTM) condition.

When examined individually, one expert (F1 = .55) does strictly better than LSTM (F1 is in the range[1] of [0.3558, 0.4372]), and one expert (F1 = .39) does comparably to LSTM (Figure 3). While the other two experts' rules did not have a F1 score as high as LSTM, note that the experts only spent 30 minutes on the task. Within the short amount of time, 2 out of 4 experts are able to construct a model that is comparable to or even better than LSTM.

When examined collectively, we see that as a group the experts did better than LSTM. For this analysis, we chose any 3 out of 4 experts and joined their rules as a union, resulting in 4 aggregate models. We consider such aggregation an appropriate comparison in the real world engineers often work as a team. While joining the rules as a union is a naive way of collaboration, due to overlapping tasks, it is a simple measurable metric. Figure 4 shows that 3 out of 4 aggregate models (F1 = .55 for all three) are performing better than LSTM, while the remaining (F1 = .42), is close to the peak of LSTM (F1 = .4372). On average, the F1 of experts is higher than the LSTM by a range of 35.58% to 43.72%. This indicates that, within 30 minutes, and with no interaction among subjects, 3 experts together are able to produce a white-box model that outperforms a competitive opaque ML model.

5.2 A more productive rule developing process

Individually, Figure 3 shows that 3 out of 4 subjects with the assistance of our system, are able to produce a model that generalizes better than the baseline subject with the same level of expertise. Among them, 2 subjects (F1 = .55 and F1 = .39) are doing much better than the subject without the system (F1 = .22). The recall of the three experts (.13, .25, .39) are all higher than the baseline (.11). On average, the F1 and recall of the hybrid approach are higher than that of the baseline subject by 58.75% and 88.64%, respectively. This indicates that our system helps users to quickly recognize correlations between examples and labels, allowing them to create rules that correctly classify more examples. One possible reason that one of our experts with the system (F1 = .11) did not do as well is that, she may not have enough expertise in the domain. As she later said in the post-study interview, *"I have worked with categories, but not communication specifically."* Her rules actually yield a high performance (F1 = 0.78) on the training set. This score is always

[1] A crucial hyperparameter that determines LSTM performance is the number of hidden units it contains. By varying this hyperparameter, one may learn different LSTMs that produce different test set F1.

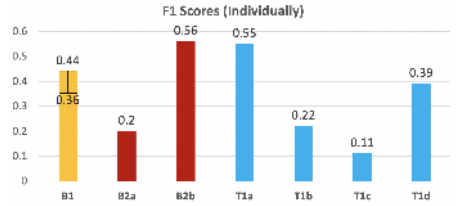

Figure 3: F1 scores comparing individual model performance of treatment group $T1$ (human-machine cooperative) subjects, against baseline $B1$ (machine-learning only) and $B2$ baseline (human only) subjects

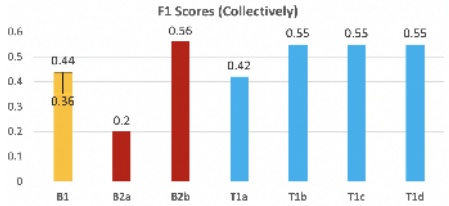

Figure 4: F1 scores comparing collective group performance of treatment group $T1$ (human-machine cooperative) subjects, aggregating all subjects except self, against baseline $B1$ (machine-learning only) and $B2$ baseline (human only) subjects

available as a reference to her. So, she might have constructed rules primarily based on the training set, which results in low performance in the test set. This shows that domain expertise is indeed necessary to be part of our model development process.

The expert with over 10 years of experience (F1 = .56) is doing better than 3 of our experts with the system. However, we are aware that this is not a fair comparison, but consider her as an upper-bound of the task. It is worth noting that one of our experts produced a collection of rules (F1 = .55) that is comparable to the super expert.

We then compared our post-hoc aggregated results to the super expert. Among the 4 aggregated rule-based models, Figure 4 shows that 3 of them yield comparable performance (F1 = .55, precision and recall vary) to that of the super expert. This shows that, with the assistance of our system, a few experts together can accomplish a task as well as a super expert, who is much more difficult and costly to find. Since combining rules as a union is only a naive way of leveraging the crowd effort, we anticipate the experts to create a better model if they collaborate in person, as opposed to offline aggregation.

5.3 An explainable model

By explainability, we mean that people can understand 'what the machine thinks', not necessarily 'how it thinks'. In the pre-study, we asked the 6 participants to write 1-2 sentences of explanation of 10 rules. To measure explainability, we treated the explanations of super expert as the ground truth, and scored others' explanations on a 5-point scale, from 1 (no match) to 5 (complete match). As a result, on average, 2 participants obtained a score of 5, suggesting that they can explain the rules as well as the super expert. 3 participants scored above 4, which shows that they can explain how each rule operates mostly correctly. The fact that experts' explanations mostly match with that of super expert also indicates that they can provide consistent explanation. After all, the rules are more intuitive and interpretable than the contemporary powerful machine learning models that can have millions of parameters.

5.4 Interviews

At the end of the study we interviewed subjects on the explainability of rules, usability of the tool, and if and how they would use the tool in their work. Most subjects expressed that the semantics of the rules made sense, given its simple disjunctive normal form and use of semantic linguistic structures. As for the machine-generated rules, a few subjects said that some predicates were surprising given their domain knowledge but noteworthy to check out: "*I was surprised to see mood imperative with a verb, which we would have not thought.*" Quantitative measures played a key rule building trust with the model, as one subject put it: "*To me trust is numbers!*" Others seemed to rely on examples: "*Looking at the examples, verifying what they were bringing, assuring that it is a good rule.*"

The tool was found to be very useful by many of our subjects, especially given that current practice is very manual. In particular, many liked interactive calculation of precision and recall in response to their selection: "*It is great to see just removing it and putting in something else and find the delta so easily.*" Another suggested this sheds off quite a bit of time from their practice: "*Something that I can complete in half a day, here it would be done in a couple of minutes.*"

All participants said that they would incorporate this tool into their practice. Some focused on initial use when they receive new data to improve models, others suggested they would use it to explore predicates: "*Looking at overall (common) predicates gave me a good idea of what I might add.*"

6 CONCLUSION

Attaining the right level of abstraction is crucial for achieving meaningful and scalable interaction in human-machine cooperative model development. Our preliminary user study suggests that when rules recommended by ML are explainable and interactive, co-created models can be more generalizable and could improve individual and collective performance on a text classification task. Much remains to be examined, but we hope our tool will guide and inform future research directions.

REFERENCES

[1] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S. Weld, Walter S. Lasecki, and Horvitz Eric. 2019. Updates in Human-AI Teams: Understanding and Addressing the Performance/Compatibility Tradeoff *(AAAI'19)*.
[2] Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick R. Reiss, and Shivakumar Vaithyanathan. 2010. SystemT: An Algebraic Approach to Declarative Information Extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*. 128–137.
[3] Richard Evans and Edward Grefenstette. 2018. Learning Explanatory Rules from Noisy Data. *JAIR* (2018).
[4] Dan Jurafsky and James H. Martin. 2014. *Speech and language processing*. Vol. 3. Prentice Hall, Pearson Education International.
[5] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *Proceedings of the*

2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 5686–5697. https://doi.org/10.1145/2858036.2858529

[6] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15)*. ACM, New York, NY, USA, 126–137. https://doi.org/10.1145/2678025.2701399

[7] Diego Marcheggiani, Michael Roth, Ivan Titov, and Benjamin Van Durme. 2017. Semantic Role Labeling. In *EMNLP*.

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[9] Jenna Wiens, John V. Guttag, and Eric Horvitz. 2012. Patient Risk Stratification for Hospital-associated C. Diff As a Time-series Classification Task. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*. Curran Associates Inc., USA, 467–475. http://dl.acm.org/citation.cfm?id=2999134.2999187

[10] Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *NIPS*.