PreCall: A Visual Interface for Threshold Optimization in ML Model Selection

Christoph Kinkeldey
christoph.kinkeldey@fu-berlin.de
Human-Centered Computing, Freie Universität Berlin
Berlin, Germany

Claudia Müller-Birn
clmb@inf.fu-berlin.de
Human-Centered Computing, Freie Universität Berlin
Berlin, Germany

Tom Gülenman
tom.guelenman@fu-berlin.de
Human-Centered Computing, Freie Universität Berlin
Berlin, Germany

Jesse Josua Benjamin
jesse.benjamin@fu-berlin.de
Human-Centered Computing, Freie Universität Berlin
Berlin, Germany

Aaron Halfaker
ahalfaker@wikimedia.org
Wikimedia Foundation
San Francisco, CA, USA

ABSTRACT

Machine learning systems are ubiquitous in various kinds of digital applications and have a huge impact on our everyday life. But a lack of explainability and interpretability of such systems hinders

HCML Perspectives Workshop, May 04, 2019, Glasgow

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

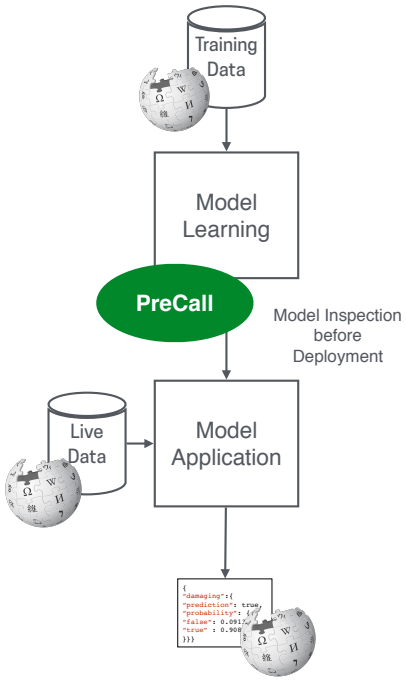

Figure 1: The PreCall approach can be localized at the deployment stage of a machine learning model.

meaningful participation by people, especially by those without a technical background. Interactive visual interfaces (e.g., providing means for manipulating parameters in the user interface) can help tackle this challenge. In this position paper we present PreCall, an interactive visual interface for ORES, a machine learning-based web service for Wikimedia projects such as Wikipedia. While ORES can be used for a number of settings, it can be challenging to translate requirements from the application domain into formal parameter sets needed to configure the ORES models. Assisting Wikipedia editors in finding damaging edits, for example, can be realized at various stages of automatization, which might impact the precision of the applied model. Our prototype PreCall attempts to close this translation gap by interactively visualizing the relationship between major model parameters (recall, precision, false positive rate and the threshold between valuable and damaging edits). Furthermore, PreCall visualizes the probable results for the current parameter set to improve the human's understanding of the relationship between parameters and outcome when using ORES. We describe PreCall's components and present a use case that highlights the benefits of our approach. Finally, we pose further research questions we would like to discuss during the workshop.

INTRODUCTION

With rising concerns over the utilization of machine learning (ML) algorithms in everyday activities as well as high-stakes environments such as the law [3], new directions for the development and deployment of algorithmic systems for non-technical groups have emerged. On the one hand, legislative imperatives such as the GDPR's implicit right to explanation [9] have led to design strategies that supply explanations for ML algorithms in user interfaces. However, these are predominantly created by technical experts and can prove unsuitable for non-technical groups due to their formal commitments [5, 8]. As an extension to explanations, interactive approaches, such as sample review, feedback assignment, model inspection, and task overview, have been suggested in order to foster more meaningful participation in systems that use ML algorithms [4]. We see these interactive approaches as integral to the goal of making ML more accessible. In our position paper, we focus on the model inspection facet of a machine learning system (cp. Figure 1), and present an interactive user interface for the ML back-end service ORES.

USE CASE: ORES

Only a few years after its inception, the number of active volunteers in Wikipedia grew exponentially. At the same time, this success leads to increasing vandalism in Wikipedia. The English Wikipedia, for example, receives over 150 thousand new edits every day, which go live immediately and without verification. Wikipedians accept this risk of an open encyclopedia but work tirelessly to maintain quality. However, this was no longer possible manually from this point on. Due to its ongoing growth, Wikipedia entered into a phase of automation, and many quality control tools, such as

[1]ClueBot NG pre-classifies edits in Wikipedia by with a Bayesian Classifiers to reduce the percentage of false positives. Then an artificial neural network is used to classify the detected vandalism. ClueBot NG generates a vandalism probability for each edit.

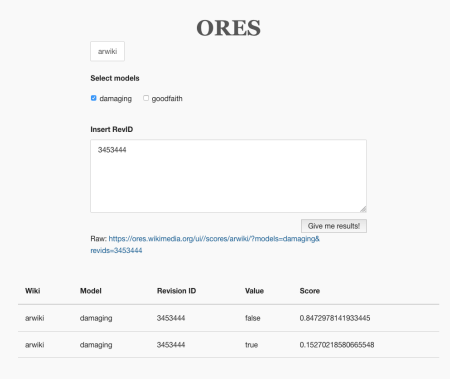

Figure 2: Current interface for the ORES damaging detection system, (https://ores.wikimedia.org/ui/**).**

ClueBot NG[1] were developed. However, developers in Wikipedia are volunteers who often learn Javascript/Python by themselves. They are non-technical experts and do not have deep enough technical expertise in machine learning terms and practices, and therefore, they lack the expertise to develop the machine prediction models necessary to power quality control tools. The Wikimedia Scoring Platform team [1] tackled this challenge and had developed ORES, a machine prediction service developed and maintained by professionals, but intended to be used by volunteer tool developers.

As of now, ORES offers a web API to make use of its models, and a very simple user interface exists which allows people to retrieve scoring information about edits across a multitude of wikis (Figure 2). ORES is a back-end service that allows other tools to simply provide one or more revision ID(s) and receive the probability scores (ranging from 0 to 1) for the respective revision(s) as being “damaging” or ”not damaging” as output.

Developers who want to apply the ORES damaging prediction need to choose a threshold of confidence that supports the work practices they are designing for. By choosing a low threshold, recall is increased at the cost of precision. By choosing a high threshold, precision is increased at the cost of recall. Thus these formal ML terminology of "precision" and "recall" are not common knowledge, so inspecting the model and determining appropriate parameters for a specific purpose is not well supported for these non-ML experienced developers. Thus, in the next section, we describe existing challenges that occur when employing ORES as quality control system.

HUMAN-CENTERED PARAMETER OPTIMIZATION

Halfaker et al. [6] describe the case of PatruBot from Spanish Wikipedia. An editor developed PatruBot by using ORES to revert damaging edits in Spanish Wikipedia automatically. However, soon after its initiation, the Wikimedia Scoring Platform team received much feedback from editors who did not understand why PatruBot reverted their edits. After interrogation, it occurred that the bot reverted edits that passed a low threshold likelihood of being damaging. In case of a fully automated quality control process, the model needs to be optimized to a high *precision*, i.e., only damaging edits are flagged, which results in a lower *recall*, i.e., some suspicious edits remain undetected.

What we derive from this case is that even with knowledge about ORES, it is not straightforward for people to come up with a confidence threshold that meets their operational requirements (high precision at the cost of recall). The interplay of model fitness statistics/expectations requires interpretation on a case-by-case basis. Our approach is to support this tool developer understanding and the development of an interpretive process using an interactive visual user interface.

Therefore, we see this use case as a particularly fruitful setting for developing interfaces that lower the barrier for non-technical community access to ML. Accordingly, we were motivated to prototype PreCall, an interactive visual interface to guide non-technical experts, i.e., editors, to develop a mental model of the functioning of ORES when selecting suitable parameters for their model application.

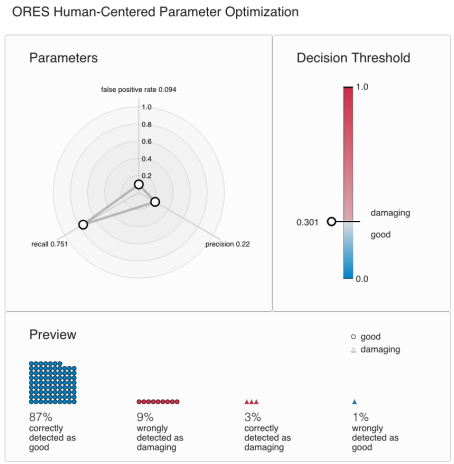

Figure 3: Interactive user interface of PreCall consisting of a parameter view (upper left), decision threshold slider (upper right), and result preview (bottom).

The interplay and the actual effect of different parameter sets on the outcome of ORES are challenging to understand for non-technical experts. Previous research has shown that interactive visualizations that enable people to adjust model parameters help them to make more effective use of ML-services [2, 7]. In our research, we build upon this line of research and seek to support people in finding optimal parameters that meet their requirements, without having to understand how exactly the system works internally.

THE PRECALL VISUAL INTERFACE DESIGN

The visual interface aims to inform a person about possible model parameters of the damaging model, and to support the interpretation of the outcome expected from each parameter set. Ultimately, the interface is supposed to provide support in finding the right threshold to decide at what score edits are being marked as damaging in a way that represents a fitting/suitable combination of parameters tailored to the requirements.

Parameter Space

As ORES users may interpret their desired outcome from a parameter- or threshold-perspective, the first goal was to show the relationship of the three major parameters of the ORES damaging model: recall, precision, and false-positive rate. In the GUI they are represented as three axes of a radar chart (Figure 3, top left view). A person can vary any parameter and the other two are updated instantly. The next aim was to demonstrate how the damaging threshold relates to the model parameters. A slider next to the radar chart represents the threshold resulting from the selected parameters (Figure 3, top right view). A color gradient illustrates the fact that the transition from good to damaging edits is fluid, i.e., there is a range of uncertainty. Changing the threshold in the slider also immediately changes the values in the shown parameter set of the radar chart. In this way, interactions facilitate the exploration of different thresholds and model parameters, as well as their interdependence.

Preview of Results

Another crucial goal of PreCall is to demonstrate how the outcome of the model changes with different parameters. The view on the bottom (Figure 3, “Preview”) shows the predicted outcome for the chosen parameters as stacked symbols. This view is designed to provide an intuitive representation of the expected result to let the user quickly grasp the number of elements belonging to the different groups: true negative, false positive, true positive, and false negative flags of edits. Color expresses how the algorithm tagged the edits: good (blue) and damaging (red). The shape of the elements represents their true state: good (circle) and actually damaging (triangle) edits. Compared to the common way of showing classification results in a confusion matrix (e.g. [7]), we assume that this visualization provides a more intuitive representation of the outcome of a parameter set. Moreover, by also adapting

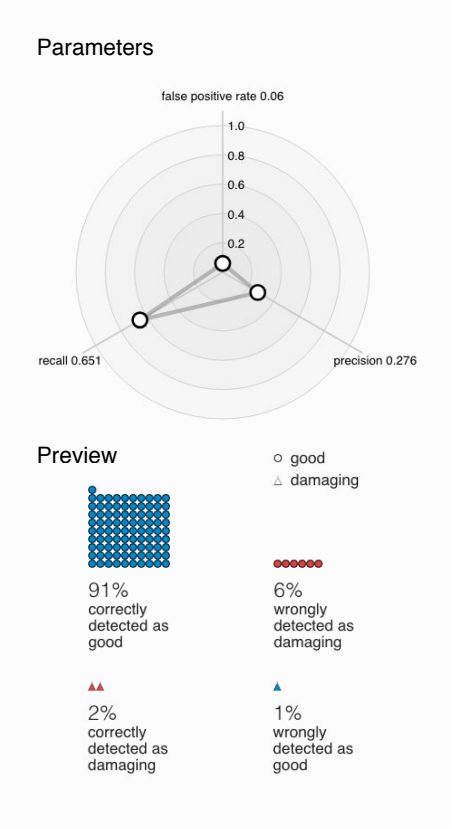

Figure 4: The result for the use case of semi-automated edit review: a threshold of 0.4 minimizes the number of detected damaging edits.

instantly, this preview further strengthens PreCall's interpretive support by suggesting a possible future application of current settings.

Determining the Parameter Set for Semi-Automated Edit Review

We use the second scenario, the semi-automated review of edits described above, as a demonstration of how PreCall can help finding the optimal parameter set for a certain application:

(1) We start with a threshold of 0.5, which results in recall of 0.569, precision of 0.347, and false positive rate of 0.038. With this parameter set, the number of falsely detected good edits is still quite high ("2% wrongly detected as good"), as we have the same amount of correctly detected damaging edits.
(2) In order to let the system find more damaging edits, we decrease the decision threshold to 0.3. The parameter view reveals that recall goes up (0.751) and precision down (0.22). The fraction of "wrongly detected as good" edits went down to 1%, however there are still 12% of edits altogether that are (correctly and falsely) detected as damaging and have to be reviewed manually.
(3) Trying out other thresholds, we find a better choice: with a threshold of 0.4 the number of edits that are detected as damaging is minimized to 8% (with 6% wrongly and 2% correctly detected). We can interpret the preview of the outcome as an acceptable payoff, given our purpose of reviewing a small number of uncertain edits among a large set of edits.
(4) With this understanding, we are satisfied with this parameter set and use the chosen parameters to check new data for damaging edits.

This scenario shows how PreCall, with its integrated visual approach, is intended to support the configuration of the ORES damaging model. We hope we can show in planned user studies that PreCall helps people build a meaningful understanding of the parameters, their relationship, and how they affect the possible outcome.

DISCUSSION

In this position paper we described the context, the requirements, and the current design rationales of the work-in-progress development of PreCall. The main goal of the approach is to support the editors in Wikimedia projects, i.e. non-technical experts, in arriving at a case-specific meaningful interpretation when selecting parameters for the ORES damaging model. The current prototype serves as a demonstration of the concept and as testing platform for the wider community. To explore the suitability of PreCall to support the interpretation of parameter selection for specific case-by-case usage of ORES, we envision a qualitative user study with Wikipedia editors. A particular concern is the level of abstraction PreCall should provide, such as whether our inclusion of measures like precision and recall is interpretable for Wikipedia tool developers. Therefore, our study should also compare ours

to more abstract approaches such as an interactive confusion matrix as proposed by Kapoor et al. [7]. Another possible qualitative dimension to our studies is comparing the understanding gained by using PreCall as opposed to reading the officially supplied documentation for ORES parameters (e.g., https://www.mediawiki.org/wiki/ORES/Thresholds). If our approach turns out to be useful, a future goal would be to provide the Wikimedia community with an enhanced version of PreCall for long-term field studies. In this way, we hope to improve our understanding of how such visual interfaces can impact the acceptance and usage rate of ML-systems in the community. In this workshop we would like to raise the following topics. We see visual parameter selection support approaches like PreCall as valuable contributions to participatory machine learning. Our strategy is to facilitate better understanding of machine learning systems without necessarily pursuing the goal of making them entirely transparent. We are convinced that following this strategy, visual approaches have the potential to foster a better understanding of machine learning-based decision making.

REFERENCES

[1] [n. d.]. Wikimedia Scoring Platform Team. https://www.mediawiki.org/wiki/Wikimedia_Scoring_Platform_team. Accessed: 2019-02-07.

[2] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 337–346.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[4] John Dudley and Per Ola Kristensson. 2018. A Review of User Interface Design for Interactive Machine Learning. (March 2018). https://doi.org/10.17863/CAM.21110

[5] Lilian Edwards and Michael Veale. 2017. *Slave to the Algorithm? Why a 'Right to an Explanation' Is Probably Not the Remedy You Are Looking For*. SSRN Scholarly Paper ID 2972855. Social Science Research Network, Rochester, NY. https://papers.ssrn.com/abstract=2972855

[6] Aaron Halfaker, R Stuart Giger, Jonathan T Morgan, Amir Sarabadani, and Adam Wight. 2018. ORES: Facilitating re-mediation of Wikipedia's socio-technical problems. (2018).

[7] Ashish Kapoor, Bongshin Lee, Desney S Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. *CHI* (2010), 1343.

[8] Tim Miller, Piers Howe, and Liz Sonenberg. 2017. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences. *arXiv:1712.00547 [cs]* (Dec. 2017). http://arxiv.org/abs/1712.00547 arXiv: 1712.00547.

[9] Andrew Selbst and Julia Powles. 2018. "Meaningful Information" and the Right to Explanation. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research)*, Sorelle A. Friedler and Christo Wilson (Eds.), Vol. 81. PMLR, New York, NY, USA, 48–48. http://proceedings.mlr.press/v81/selbst18a.html