AI Evaluation Authorities: A Case Study Mapping Model Audits to Persistent Standards

Arihant Chadda[1*], **Sean McGregor**[2*], **Jesse Hostetler**[2], **Andrea Brennen**[1]
[1]IQT Labs
[2]UL Digital Safety Research Institute
achadda@iqt.org, sean.mcgregor@ul.org, jesse.hostetler@ul.org, abrennen@iqt.org

Abstract

Intelligent system audits are labor-intensive assurance activities that are typically performed once and discarded, along with the opportunity to programmatically test all similar products for the market. This study illustrates how several incidents (i.e., harms) involving Named Entity Recognition (NER) can be prevented by scaling up a previously-performed audit of NER systems. The audit instrument's diagnostic capacity is maintained through a security model that protects the underlying data (i.e., addresses Goodhart's Law). An open-source evaluation infrastructure is released along with an example derived from a real-world audit that reports aggregated findings without exposing the underlying data.

Introduction

Many real-world applications of knowledge discovery, knowledge extraction, search, and computer network security involve a Named Entity Recognition (NER) step. NER is the task of recognizing a variety of "entities" within text. For example, the text "2012's [DATE] AlexNet [PRODUCT] is named for Alex Krizhevsky [PERSON]," has three entity types for dates, products, and persons.

Likely examples of failed NER appear frequently in the AI Incident Database (AIID) of (McGregor 2021), which catalogs examples of AI harms produced in the real world. Though not referenced explicitly in the incident reports, NER is a foundational Natural Language Processing (NLP) task undergirding a great many products. Most incident reports potentially related to NER center on the user-facing issues of the technologies, including AI incidents 317 (Bug in Facebook's Anti-Spam Filter Allegedly Blocked Legitimate Posts about COVID-19), 363 (Facebook's Automated Moderation Mistakenly Flagged Landmark's Name as Offensive), and 392 (Facebook's AI-Supported Moderation Failed to Classify Terrorist Content in East African Languages) (Dickinson 2020; Lam 2021, 2015).

In a typical real-world deployed Machine Learning (ML) system, a NER model would be only one of many subsystems and models that are composed in various ways and wrapped by a variety of user interfaces to facilitate the system's overall use case. This complexity obfuscates the explicit role that an NER model plays in any incidents produced by the system, but this is standard practice when it comes to moving from Research and Development (R&D) to end-user-facing production. Still, the frequency of incidents that likely link to NER models underscores the importance of the NER task and the need to mitigate incidents arising from it. Any multi-component system can fail if an individual component produces erroneous or harmful outputs.

While the specific issues leading to these incidents cannot be localized without proprietary knowledge of Meta's implementation, the incidents in Table 1, are similar and involve probable NER failures on Meta's Facebook platform.

IQT Labs[1] has conducted several audits where we assessed the safety and fairness properties of AI tools and systems (Brennen and Ashley 2021; Brennen et al. 2022; Ashley et al. 2023). The current paper focuses on our audit of the RoBERTa model (Liu et al. 2019) and variants thereof (Conneau et al. 2019), which are pre-trained LLM architectures we audited over several months. The variants audited included RoBERTa-base, RoBERTa-large, XLM-RoBERTa-base, and XLM-RoBERTa-large, which collectively were downloaded about 27.2M times on HuggingFace between July 15th and August 15th of 2023 (HuggingFace 2022). As part of that audit, we developed a multilingual NER programmatic assessment that exposed model limitations and identified a model attack surface component (Calix et al. 2022).

Here we extend our prior work on NER auditing by adding reproducibility and scalability to the programmatic assessment—a nontrivial exercise to develop and implement an applied framework for reproducible model assessment. We call such a framework an "Evaluation Authority":

Definition 1 ***Evaluation Authority***. *A programmatic and secured instantiation of one or more tests maintained by a trusted organization for the purpose of establishing and iterating safety standards and/or scores.*

While providing specific insights into bias, ethics, security, and user experience risks, each of our previous audits were applied only once to single systems under test and took several months per system. Reproducibility and scalability

*These authors contributed equally.

[1]IQT Labs is a subsidiary of IQT, the not-for-profit strategic investor for the U.S. national security community and its allies, that explores proxy problems of national interest in the open source.

Incident #	Incident Title	Speculated NER Cause
317	Bug in Facebook's Anti-Spam Filter Allegedly Blocked Legitimate Posts about COVID-19	Entity-specific post blocking suggests a possible failure somewhere along the NER chain, which prevented accurate resolution of well-known news sources like Business Insider and The Atlantic
363	Facebook's Automated Moderation Mistakenly Flagged Landmark's Name as Offensive	Likely the NER system either did not correctly resolve Plymouth Hoe as a famous landmark in the United Kingdom or was incorrectly overruled by a keyword detection model in Facebook's ensemble
392	Facebook's AI-Supported Moderation Failed to Classify Terrorist Content in East African Languages	Postings in Arabic, Somali, and Kiswahili that included propaganda about Harakaat al-Shabaab al-Mujahideen (al-Shabaab), an African terrorist organization, were not flagged by the platform, suggesting failed entity resolution

Table 1: AIID incidents 317, 363, and 392 all involve likely NER issues on the Facebook platform.

are essential to keep pace with the development iterations of AI tools, where variants of one architecture are downloaded over 27M times in a month. The Evaluation Authority approach transitions audit outputs from one-time assessments to standards characterizing the entire product category.

Many aspects of AI assurance are built around qualitative processes, making it difficult to propose effective modes of programmatic testing. While harms like those in the AIID can be illustrative, they suffer from the availability heuristic—a proclivity toward recording the most readily observable harms rather than the most important. However, the propensity to produce previously-experienced harms *can* be tested programmatically with ecological validity.

The key property of the Evaluation Authority concept is its security model. We contrast Evaluation Authorities, which are used for *system assessment*, with *benchmarks*, which are evaluations conducted for the purpose of *system improvement*. The key difference is that benchmarks are optimization targets, and are thus subject to Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure" (Strathern 1997). Russell and Norvig's definitive textbook of AI (Russell and Norvig 2009) succinctly describes how to avoid this hazard:

> ...really hold the test set out—lock it away until you are completely done with learning and simply wish to obtain an independent evaluation of the final hypothesis. (And then, if you don't like the results ... you have to obtain, and lock away, a completely new test set if you want to go back and find a better hypothesis.)

Restricting access to test data in this way is essential to preserving its validity. Yet in our experience evaluating commercial solutions, this principal is frequently forgotten beyond the classroom. We thus designed a software infrastructure for Evaluation Authorities that allows for critical safety property assessment while defending the integrity and diagnostic utility of the test data for safety use cases.

Once incorporated into a public Evaluation Authority, an audit methodology transitions from a singular evaluation exercise to a collection of operational requirements a product must satisfy before it can be responsibly deployed. Evaluation Authorities provide a clear, public assessment that helps interested stakeholders determine the deployment circumstances for which model deployment is unsafe, including the propensity to produce incident recurrence. In doing this openly, the Evaluation Authority also establishes product standards that market entrants can be expected to clear.

In a rapidly developing world, such standards can easily become outdated as the world or the nature of the models shift through time. Consequently, we emphasize evaluations that can be extended through time as new incidents are indexed and subject matter expertise advances the safety case. As an initial step, our tests focus on a narrow scope: do Large Language Models (LLMs) applied for multilingual NER tasks exhibit language-related biases? As we expand the scope of these tests to more potential NER incidents, we (or a competing NER standard-by-assessment) gain insight into broader classes of problems and provide a capacity to notify system deployers of their system risks.

The basic Evaluation Authority infrastructure can be extended to any digital system type or task, provided that data associated with incidents can be captured, synthesized, or mocked. To stress test our open-source[2] Evaluation Authority infrastructure, we also performed large-scale testing across a variety of image classifiers to assess their susceptibility to common data corruptions.

The contributions of this work to reduce the likelihood of incident recurrence are (1) repackaging a laboriously produced audit as a product standard that scales to an entire product category; (2) open sourced infrastructure-as-code enabling auditors to stand up their own Evaluation Authorities; (3) detailing of a best practice wherein assessment data is protected from Goodhart's Law and managed by those organizations concerned with maintaining the marketability of their standards; and (4) demonstrating how an Evaluation Authority for NER multilingual robustness assessment has the capacity to identify upstream issues related to three incidents archived in the AIID.

Background & Motivation

Machine learning-based solutions are commonly iterated and improved in production (Khlaaf 2023), but incident data critical to the assessing a propensity to harm is rarely elevated to the same stature of transportation and medical incident data. A typical **AI Incident Cycle** presents as follows:

1. The AI system is made available to users after limited in-house testing

[2]Infrastructure available at https://dyff.io

2. An AI incident is reported
3. A remediation is proposed and the system is updated and/or taken offline
4. An updated system is re-deployed to users

The associated incident data may be added to the next model training run, but it is rarely analyzed and incorporated into future risk assessments, blinding the developers from insights into the safety properties of successive model generations. How can the company know when the underlying problem is solved? Or phrased differently: how can similar AI incidents be prevented in the future?

The Missing Translational Layer

Several public and private sector frameworks and guidelines attempt to address these questions, but none provide the translational layer between strategic high-level issues and tactical, discrete incident resolution (ICO 2020; Google 2018; ODNI 2020). Frameworks and guidelines that describe what to test without describing how to test it open the door for exercises in superficial box-checking.

Writing a test is also non-trivial. Through our audit of RoBERTa, we understood the importance of taking a task- or use-case-centric approach to auditing, as it allows for meaningful comparisons while minimizing misrepresentation of conclusions (Brennen et al. 2022). The ability to evaluate a model is predicated upon having data that expresses the phenomena to be tested, meaning datasets and assessments are coupled. To this end, testing faces a bootstrapping problem where the lack of high-quality datasets for AI-Governance-related assessments significantly limits what can be tested (McGregor and Hostetler 2023). AI incident remediation efforts and audits of AI systems provide an opportunity to produce safety datasets. Contributing them to a public Evaluation Authority like the one we implemented in this paper now allows for regression testing or comparing the propensity for different models to produce incidents.

These datasets, though, need to be kept secret from organizations whose products are under test. Specifically, the data itself should never be exposed, and only high-level findings should be disclosed to make black-box optimization against the test difficult and thus preserve the validity of the assessment over the long term. This protection encourages product creators to focus on remediating the high-level issue rather than attempting to overfit the assessment data. While this may obscure specific failure modes of a model, it inspires a virtuous development mindset that looks to minimize harms instead of discrete failure instances.

It is clear from observation of current AI deployments that the current AI model deployment workflow is inadequate, especially for models deployed in high-stakes scenarios. Furthermore, framing it around AI incident response and mitigation provides a forcing mechanism for the industry to learn from its mistakes and the mistakes of others. While an Evaluation Authority will not solve all the issues, it provides a foundation to build and iterate on for a better functioning AI safety culture. An Evaluation Authority would serve as a release gate for the model, and as both evaluations and models would be periodically enhanced, the model would be re-evaluated as part of the "AI Incident Cycle".

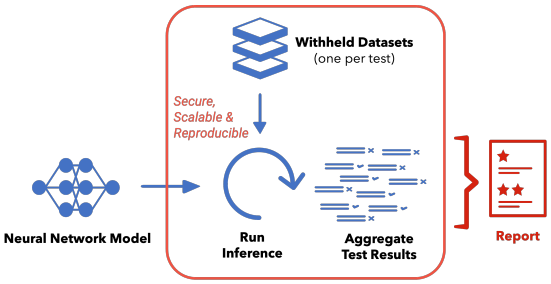


Figure 1: Evaluation Authority input and output diagram with the four components: models, datasets, and tests that aggregate into a report.

The demonstration described in this paper is meant to concretize the development and utility of an Evaluation Authority as the missing translational layer. We focus on NER to narrow test creation to a specific use case. Instead of attempting to address an ill-defined problem like testing an LLM for "bias," we operationalize one specific aspect of bias into something that can be programatically and quantitatively assessed. In the current paper, we focus on multilingual robustness for NER, defined as differences in performance between sets of person names that are common in different languages. The corresponding multilingual names dataset is maintained within the Evaluation Authority infrastructure and is not exposed to the user. The dataset[s] and assessment[s] inference code sit within the infrastructure-as-code implementation that defines an Evaluation Authority.

Evaluation Authority Implementation

At a high level, the Evaluation Authority has three inputs: the system under test, protected datasets, and system assessments (Figure 1). The system under test is provided by the system owner and uploaded into the Evaluation Authority. Datasets are protected by running them through the system under test inside the Evaluation Authority, where the system owner cannot access them. Finally, the raw results are aggregated to assemble a report of the system's performance.

When building the Evaluation Authority, there was a concerted effort to use canonical ML tools to lower the cognitive load of using and contributing. This can be seen in the block diagram in Figure 2.

The Evaluation Authority is composed of three layers: Developer, Auditor, and Reporting.

The Developer Layer targets Data Scientists and ML Engineers—those who train models and create the housing around them. In AI incident response or AI model auditing, this is the group that created and deployed the production model that caused the incident. The Developer Layer offers two entry points into the platform, either selecting an LLM from HuggingFace Hub or supplying a packaged model in the BentoML format (HuggingFace 2022; Yang et al. 2022). This allows both open- and closed-source models to be used with the Evaluation Authority. Developers can utilize assess-

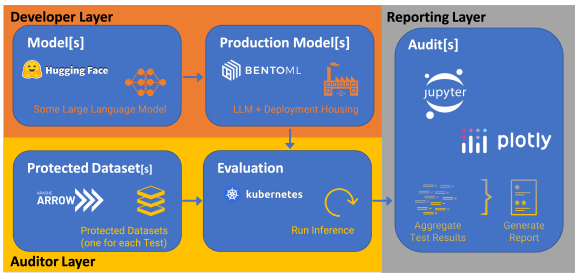
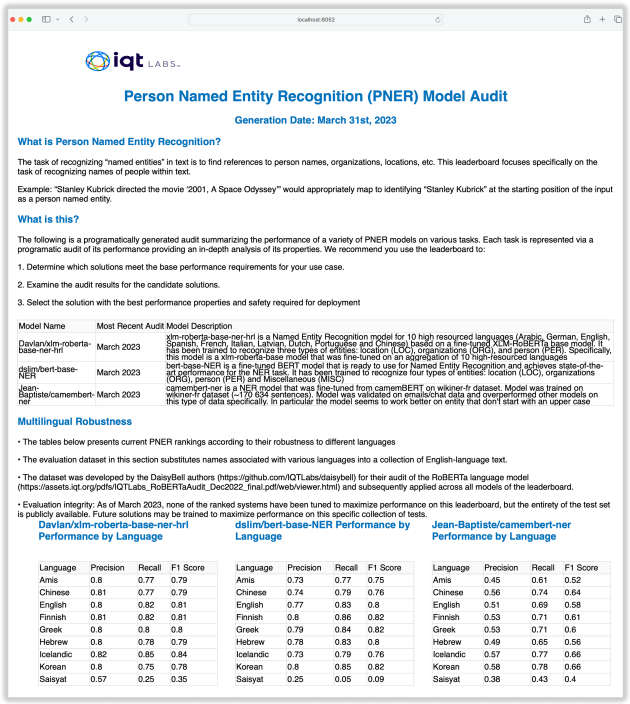


Figure 2: Block diagram with the open-source software tools that compose the Evaluation Authority infrastructure.

ments that already exist in the Evaluation Authority instantiation and/or write their own.

The Auditor Layer is where tests are written and run. Feasibly, these are generated during AI model auditing and AI incident response. To run tests in a reasonable amount of time, the auditing and response teams would need to create some version of the infrastructure. Given that it already exists in the Evaluation Authority openly, it makes sense to build on it instead. Developers can be auditors, but they do not have to be. In some cases, there are benefits to having the same people who created the system write relevant tests (i.e., cost benefits), but they are also most biased toward the capabilities of a system. In these cases, outside auditors may be better positioned to identify limitations. Each test is coupled with a protected dataset—in order to prevent overfitting; these datasets are not publicly released. The hope is that this inspires model builders to address the root cause of a performance problem instead of training on the specific examples that are part of the test. Datasets are spooled from cloud storage using Apache Arrow, and assessment inference sessions are parallelized using Kubernetes (The Apache Software Foundation 2023; The Kubernetes Authors 2023). The distributed assessment infrastructure allows for the scaling required to run any assessment on any model.

Finally, the Reporting Layer compiles and visualizes results. This report is the output of the Evaluation Authority for a typical user and is what would be interpreted by the AI model auditing and AI incident response teams. We built versions in Jupyter Notebooks and Plotly Dash (Kluyver et al. 2016; Plotly Technologies Inc. 2015). While tests can be run on a single model, the comparative analysis enabled by running the same test on multiple models allows for relative benchmarking. Given that for the majority of AI-Assurance-related issues, it is difficult to prescribe a defensible threshold of "good enough" for any assessment, we hope that relative benchmarking will inspire incremental improvement efforts. Much like benchmarking on task-specific datasets like Microsoft Common Objects in Context or ImageNet have inspired revolutions in object detection and image classification, respectively, we hope commonly-accessible safety assessments will do the same to safety and fairness (Deng et al. 2009; Lin et al. 2015). The goal is to inspire commercial actors to compete to produce the safest models. This, in turn, will mitigate future AI incidents.



iqt LABS

Person Named Entity Recognition (PNER) Model Audit

Generation Date: March 31st, 2023

What is Person Named Entity Recognition?

The task of recognizing "named entities" in text is to find references to person names, organizations, locations, etc. This leaderboard focuses specifically on the task of recognizing names of people within text.

Example: "Stanley Kubrick directed the movie '2001, A Space Odyssey'" would appropriately map to identifying "Stanley Kubrick" at the starting position of the input as a person named entity.

What is this?

The following is a programatically generated audit summarizing the performance of a variety of PNER models on various tasks. Each task is represented via a programatic audit of its performance providing an in-depth analysis of its properties. We recommend you use the leaderboard to:

1. Determine which solutions meet the base performance requirements for your use case.
2. Examine the audit results for the candidate solutions.
3. Select the solution with the best performance properties and safety required for deployment

Model Name	Most Recent Audit	Model Description
Davlan/xlm-roberta-base-ner-hrl	March 2023	xlm-roberta-base-ner-hrl is a Named Entity Recognition model for 10 high resourced languages (Arabic, German, English, Spanish, French, Italian, Latvian, Dutch, Portuguese and Chinese) based on a fine-tuned XLM-RoBERTa base model. It has been trained to recognize three types of entities: location (LOC), organizations (ORG), and person (PER). Specifically, this model is a xlm-roberta-base model that was fine-tuned on an aggregation of 10 high-resourced languages
dslim/bert-base-NER	March 2023	bert-base-NER is a fine-tuned BERT model that is ready to use for Named Entity Recognition and achieves state-of-the-art performance for the NER task. It has been trained to recognize four types of entities: location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC)
Jean-Baptiste/camembert-ner	March 2023	camembert-ner is a NER model that was fine-tuned from camemBERT on wikiner-fr dataset. Model was trained on wikiner-fr dataset (~170 634 sentences). Model was validated on emails/chat data and overperformed other models on this type of data specifically. In particular the model seems to work better on entity that don't start with an upper case

Multilingual Robustness

- The tables below presents current PNER rankings according to their robustness to different languages
- The evaluation dataset in this section substitutes names associated with various languages into a collection of English-language text.
- The dataset was developed by the DaisyBell authors (https://github.com/IQTLabs/daisybell) for their audit of the RoBERTa language model (https://assets.iqt.org/pdfs/IQTLabs_RoBERTaAudit_Dec2022_final.pdf/web/viewer.html) and subsequently applied across all models of the leaderboard.
- Evaluation integrity: As of March 2023, none of the ranked systems have been tuned to maximize performance on this leaderboard, but the entirety of the test set is publicly available. Future solutions may be trained to maximize performance on this specific collection of tests.

Davlan/xlm-roberta-base-ner-hrl Performance by Language

Language	Precision	Recall	F1 Score
Amis	0.8	0.77	0.79
Chinese	0.81	0.77	0.79
English	0.8	0.82	0.81
Finnish	0.81	0.82	0.81
Greek	0.8	0.8	0.8
Hebrew	0.8	0.78	0.79
Icelandic	0.82	0.85	0.84
Korean	0.8	0.75	0.78
Saisyat	0.57	0.25	0.35

dslim/bert-base-NER Performance by Language

Language	Precision	Recall	F1 Score
Amis	0.73	0.77	0.75
Chinese	0.74	0.79	0.76
English	0.77	0.83	0.8
Finnish	0.8	0.86	0.82
Greek	0.79	0.84	0.82
Hebrew	0.78	0.83	0.8
Icelandic	0.73	0.79	0.76
Korean	0.8	0.85	0.82
Saisyat	0.25	0.05	0.09

Jean-Baptiste/camembert-ner Performance by Language

Language	Precision	Recall	F1 Score
Amis	0.45	0.61	0.52
Chinese	0.56	0.74	0.64
English	0.51	0.69	0.58
Finnish	0.53	0.71	0.61
Greek	0.53	0.71	0.6
Hebrew	0.49	0.65	0.56
Icelandic	0.57	0.77	0.66
Korean	0.58	0.78	0.66
Saisyat	0.38	0.43	0.4

Figure 3: Audit report compiled by the Evaluation Authority for three LLMs compared on the task of NER[3].

NER Evaluation Authority Report

As an example, we generate a report[3] for a side-by-side comparison of three LLMs on the multilingual robustness NER assessment we replicated. All three models: "Davlan/xlm-roberta-base-ner-hrl," "dslim/bert-base-NER," and "Jean-Baptiste/camembert-ner" are from HuggingFace Hub and have been fine-tuned for the NER task making their comparison intuitive. These three models were chosen to determine which multilingual model was best suited for certain languages of interest.

The report contains information about the model evaluation task, context about how to use the document, and context about the models themselves. Finally, it includes results from a scalable version of the multilingual robustness assessment adapted from Calix et al. (2022) for the three models. The report represents a snapshot summary of performance and lists the date when the specific set of tests was run. This pseudo-version control enables a form of continuous assurance where a new report can be generated if a model is ever changed, but the old model can still be compared as well. The example report can be seen in Figure 3.

As the output of the Evaluation Authority, the report is meant to provide appropriate context but also leave the results up for interpretation to its consumers. Without knowing how a model is meant to be used, it would be difficult to prescribe a recommendation about which model to use, so such statements are intentionally left out of the report. Still,

[3]Full-size version hosted at https://dyff.io/blog/auditing-llms-for-multilanguage-ner

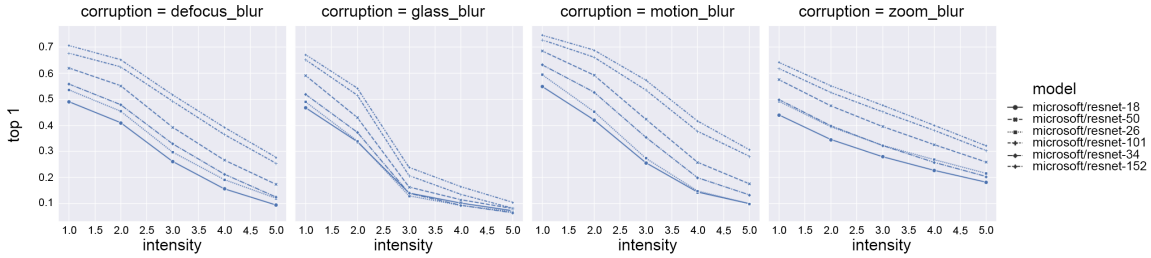


Figure 4: Degradation of ResNet model performance given blur intensity data corruption run using the Evaluation Authority.

there is immense value to practitioners and auditors when they look to understand the limitations of AI models.

How to Use the Evaluation Authority

There is both a retroactive and proactive case for the Evaluation Authority and its report output. To illustrate this, we imagine how to address an incident such as AI incident 392 by creating a new evaluation, similar to the multilingual robustness assessment, that quantifies the incident's failings.

The retroactive case begins after the AI incident is discovered, in this case, after Arabic, Somali, and Kiswahili propaganda postings about Harakaat al-Shabaab al-Mujahideen (al-Shabaab) were not flagged by the Facebook platform. The retroactive corrective action is to create a new testing dataset to evaluate the extent of the failure mode in the system. The team responsible for remediating this issue collects a dataset of posts that should have been flagged, implements new scoring metrics as appropriate, and uploads these new artifacts to the Evaluation Authority. Then, moving forward, a report on this new failure mode will be generated along with any other relevant tests already in the Evaluation Authority assessment corpus. After confirming the finding, new training efforts can also be evaluated using this test assuring the issue is acceptably benchmarked for remediation.

The proactive case begins at the model selection stage during development. If performance on Arabic, Somali, and Kiswahili names is important for the use case, as in AI incident 392, those evaluating candidate systems could either select existing tests that evaluate this performance, or author new tests and add them to the Evaluation Authority. Then, all candidate systems are subjected to the selected battery of tests and the models best suited for the use case are selected based on their relative empirical performance. In this way, the Evaluation Authority helps to de-risk models before deployment into production. Submitting a newly trained model to the Evaluation Authority provides a clear assessment according to available tests. This can highlight potential concerns, opportunities for improvement, and provide public evidence of the model creator's due diligence.

Extending the Evaluation Authority

The current prototype implementation of the Evaluation Authority contains one test for LLMs: the multilingual robustness assessment, but the Evaluation Authority infrastructure is model-agnostic and extensible. To demonstrate this, we also implemented a test for image classification models. To do so, we used a dataset of image "common corruptions" (Hendrycks and Dietterich 2018) to assess the robustness of image classifiers to various types of data manipulation.

We then evaluated six different depths of pre-trained ResNet models with the hypothetical framing of looking for the smallest model that had acceptable robustness to these corruptions (He et al. 2015). Seen in Figure 4 is a graph depicting the degradation given the intensity of the blurring for each model, a component of the overall report. This is meant to illustrate that the Evaluation Authority infrastructure is agnostic to model type and task, and that as long as a test is written in the correct format, that model can be evaluated using the Evaluation Authority. This suggests that Evaluation Authorities could become foundational utilities for AI incident remediation and prevention.

Discussion

AI incident remediation and prevention requires a translational layer between AI Governance frameworks and responsibility documents and the incidents themselves. Evaluation Authorities bridge the two by operationalizing qualitative recommendations about fairness and safety as reproducible, scalable quantitative assessments. While our work was initially motivated by a desire to scale and generalize auditing and remediation processes, we designed the Evaluation Authority infrastructure with extensibility at its core. We believe this could be a viable path forward for public, community-supported model evaluation.

If adopted, hosted, and supplemented with additional tests, we hope this approach will become standard in ML R&D and production. There are significant risks associated with deploying AI models, and too often, these risks are difficult to know, even for system developers. A public, open evaluation mechanism is an important step in identifying and mitigating many risks and future AI incidents.

For private and public organizations, assuring AI systems should be a top priority. In this paper, we have offered an example of a scalable, extensible framework that enables actionable, deliberate efforts. We encourage others to use this applied framework as a starting point—adapt it and extend it to create meaningful test harnesses for AI models.

If an Airbus plane crashes, they are not permitted to withhold critical safety data from Boeing. With a similar view for the safety of intelligent systems, safety data derived from

incidents can prevent the recurrence of harm.

Acknowledgments

Thank you to Ricardo Calix, J.J. Ben-Joseph, and Ryan Ashley for providing their multilingual robustness assessment (Calix, Ben-Joseph, and Ashley 2022).

References

Ashley, R.; Brennen, A.; Ben-Joseph, J.; Mair, E.; Gogia, M.; and Calix, R. 2023. AI Assurance Audit of SkyScan, an open source data collection & auto-labeling system. https://assets.iqt.org/pdfs/IQTLabs_SkyScanAudit_May_2023.pdf/web/viewer.html. Accessed: 2023-08-07.

Brennen, A.; and Ashley, R. 2021. AI Assurance Audit of FakeFinder, an Open-Source Deepfake Detection Tool. https://assets.iqt.org/pdfs/IQTLabs_AiA_FakeFinderAudit_DISTRO__1_.pdf/web/viewer.html. Accessed: 2023-08-07.

Brennen, A.; Ashley, R.; Calix, R.; Ben-Joseph, J.; Sieniawski, G.; and Gogia, M. 2022. AI Assurance Audit of RoBERTa, an Open source Pretrained Large Language Model. https://assets.iqt.org/pdfs/IQTLabs_RoBERTaAudit_Dec2022_final.pdf/web/viewer.html. Accessed: 2023-08-07.

Calix, R.; Ben-Joseph, J.; and Ashley, R. 2022. Daisybell GitHub. https://github.com/IQTLabs/daisybell. Accessed: 2023-08-07.

Calix, R. A.; Ben-Joseph, J.; Lopatina, N.; Ashley, R.; Gogia, M.; Sieniawski, G.; and Brennen, A. 2022. Saisiyat Is Where It Is At! Insights Into Backdoors And Debiasing Of Cross Lingual Transformers For Named Entity Recognition. In *2022 IEEE International Conference on Big Data (Big Data)*, 2940–2949.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *CoRR*, abs/1911.02116.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dickinson, I. 2020. Incident 317: Bug in Facebook's Anti-Spam Filter Allegedly Blocked Legitimate Posts about COVID-19. *AI Incident Database*.

Google. 2018. Responsiblity: Our Principles. https://ai.google/responsibility/principles/. Accessed: 2023-08-07.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Hendrycks, D.; and Dietterich, T. G. 2018. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*.

HuggingFace. 2022. HuggingFace. https://huggingface.co/{{xlm-|}roberta{-base|-large|}|}. Accessed: 2023-08-15.

ICO. 2020. Guidance on the AI auditing framework. https://ico.org.uk/media/2617219/guidance-on-the-ai-auditing-framework-draft-for-consultation.pdf. Accessed: 2023-08-07.

Khlaaf, H. 2023. Toward Comprehensive Risk Assessments and Assurance of AI-Based Systems. https://www.trailofbits.com/documents/Toward_comprehensive_risk_assessments.pdf. Accessed: 2023-08-14.

Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; and Willing, C. 2016. Jupyter Notebooks—a publishing format for reproducible computational workflows. In Loizides, F.; and Schmidt, B., eds., *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87 – 90. IOS Press.

Lam, K. 2015. Incident Number 392: Facebook's AI-Supported Moderation Failed to Classify Terrorist Content in East African Languages. *AI Incident Database*.

Lam, K. 2021. Incident Number 363: Facebook's Automated Moderation Mistakenly Flagged Landmark's Name as Offensive. *AI Incident Database*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

McGregor, S. 2021. Preventing repeated real world AI failures by cataloging incidents: The AI incident database. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15458–15463.

McGregor, S.; and Hostetler, J. 2023. Data-Centric Governance. arXiv:2302.07872.

ODNI. 2020. Artificial Intelligence Ethics Framework for the Intelligence Community. https://www.dni.gov/files/ODNI/documents/AI_Ethics_Framework_for_the_Intelligence_Community_10.pdf. Accessed: 2023-08-07.

Plotly Technologies Inc. 2015. Collaborative data science. https://plot.ly. Accessed: 2023-08-07.

Russell, S.; and Norvig, P. 2009. *Artificial Intelligence: A Modern Approach, 3rd US ed.* Prentice Hall. ISBN 0-13-604259-7.

Strathern, M. 1997. 'Improving ratings': Audit in the British University system. *European Review*, 5(3): 305–321. Publisher: Cambridge University Press.

The Apache Software Foundation. 2023. Apache Arrow. https://arrow.apache.org. Accessed: 2023-08-07.

The Kubernetes Authors. 2023. Production-Grade Container Orchestration. https://kubernetes.io. Accessed: 2023-08-07.

Yang, C.; Sheng, S.; Pham, A.; Zhao, S.; Lee, S.; Jiang, B.; Dong, F.; Guan, X.; and Ming, F. 2022. BentoML: The framework for building reliable, scalable and cost-efficient AI application. https://github.com/bentoml/bentoml. Accessed: 2023-08-07.