The Journal of Open Source Software

ICAT: The Interactive Corpus Analysis Tool

Nathan Martindale [1¶] **and Scott L. Stewart** [1]

1 Oak Ridge National Laboratory ¶ Corresponding author

DOI: 10.21105/joss.06873

Software

- Review
- Repository
- Archive

Editor: Josh Borrow

Reviewers:

- @SamHames
- @jhagerer
- @jborrow

Submitted: 21 March 2024
Published: 09 June 2025

Summary

The Interactive Corpus Analysis Tool (ICAT) is a Python library for creating dashboards to explore textual datasets and build simple binary classification models to help filter through them and focus on entries of interest. This tool uses a form of interactive machine learning (IML), a paradigm of "machine teaching" (Simard et al., 2017) that sits at the intersection of the fields of human computer interaction (HCI), visual analytics, and machine learning. The intent of ICAT is to allow subject matter experts (SME) with limited to no experience in machine learning to benefit from an iterative human-in-the-loop (HITL) approach to building their own model without needing to understand the details of the underlying algorithm. This interactivity is achieved by allowing the user to create features, label data points, and visually manipulate a representation of the features to manually cluster and investigate data, while a model is trained on the fly based on these actions. ICAT is built on top of the Panel (Holoviz, 2018) library, using a combination of Vega, a custom IPyWidget using D3, and ipyvuetify, and is intended to be used inside of a Jupyter environment.

Statement of Need

Machine teaching promises to democratize machine learning algorithms and grant non-machine-learning experts the ability to train, manipulate, and work with models themselves (Simard et al., 2017). Traditionally, the process for an SME to obtain a model that aids in their data analysis is a time consuming iterative loop: they must first communicate their problem space and data to a machine learning expert, who experiments and trains a model for the SME, who then tests it and finds any issues or insufficiently learned concepts, which must then be communicated back to the ML expert, and the iterative loop continues as such. Ideally, an effective HITL training process involves the SME more directly in the training process, dramatically speeding up this iteration loop and benefiting from the SME's implicit knowledge and experience. IML seeks to provide this process through mechanisms such as feature selection (interactive featuring) and model steering (interactive labeling) (Dudley & Kristensson, 2018).

This is a challenging space for a number of reasons. The efficacy of an IML system heavily revolves around the design of the interface itself, in addition to the underlying machine learning models and the many considerations they entail. Thus, incorporating effective user experience design principles and understanding the mental models of the users as they explore and use the interface is crucial. Both quantitative and qualitative metrics must include the human element, so any research seeking to demonstrate a measured value-add or efficacy of an IML interface must incorporate user studies (Lai et al., 2023). A positive user experience additionally constrains algorithmic design in terms of speed and efficiency–an underlying model that takes minutes to train is frustrating to interact with (Fails & Olsen, 2003). Care must be taken not to treat the user like a mechanical turk or mindless oracle for the model to endlessly query Cakmak et al. (2010).

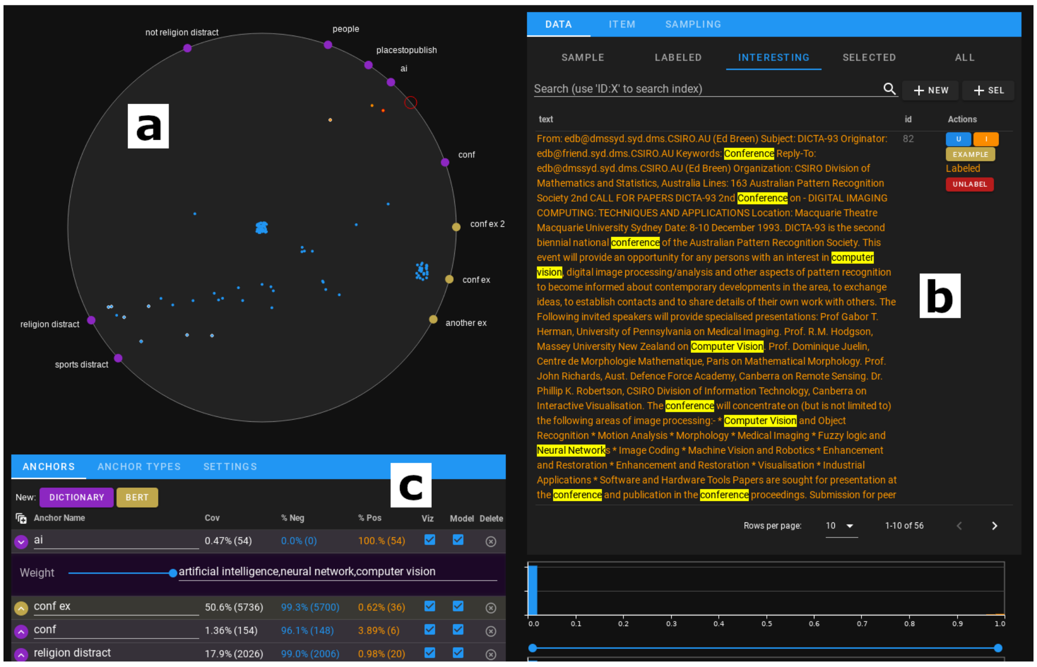
Despite these challenges, there is tremendous potential for IML to empower SMEs and allow them to benefit from the value of machine learning in their work. For the field to grow and realize this potential, a great deal more research and work are required. Our work draws heavily on the IML interface concepts proposed by Suh et al. (2019), and as of this writing there is no other open source package implementing their visuals or overall interface. ICAT seeks to fill this gap to allow other researchers to explore, build on, and compare against the concepts discussed below to further the state of the field.

Interface Concepts

The ICAT workflow trains a simple binary classification model to separate "uninteresting" from "interesting" entries in an initially unlabeled dataset. "Interesting" here is intentionally vague to refer to whatever classification signal a user is implicitly or explicitly attempting to extract. The primary use case for this is for ICAT to help filter a large collection of text objects to some smaller target subset of interest that is easier to manually review.

ICAT implements both interactive featuring and interactive labeling. Interactive featuring allows the user to create or influence the feature columns that the underlying model uses for training and predicting. In ICAT this is done through "concept anchors," or functions that return some value, nominally between 0 and 1, to represent a strength or "pull" on a provided input. An example anchor type included with ICAT is a dictionary or keyword anchor, where the feature value is a bag of words or count of the number of occurences of each keyword the user provides to that anchor. Interactive labeling is done by allowing the user to manually specify or change a label (uninteresting or interesting) for any text entry. All labeled rows are used as the training dataset for the underlying model. Once the model is "seeded," or given some minimum number of labels to train on, it begins to predict on the full dataset, and the visualizations in the dashboard are colored to reflect the interesting/uninteresting prediction for each entry.

Anchors are visualized with the AnchorViz ring (Suh et al., 2019), which are shown below in Figure 1(a). The visual representation of anchors are draggable points around the circumference of the ring, with the entries from a sample of the data rendered as the smaller points inside. Anchors pull points toward them based on the strength of influence or the magnitude of the feature value on each point. As the user drags the anchors around, the associated points then similarly move according to these varying attraction strengths, and this helps visually determine the overlap between anchors and which points are or are not represented well by the current feature set. This ability to manually position the anchors and their correpsonding points allows for various strategies for manually clustering the data, such as using anchors to pull away any distractors or known incorrect keywords from the interesting set, and so on.



text	id	Actions
From: edb@dmssyd.syd.dms.CSIRO.AU (Ed Breen) Subject: DICTA-93 Originator: edb@friend.syd.dms.CSIRO.AU Keywords: Conference Reply-To: edb@dmssyd.syd.dms.CSIRO.AU (Ed Breen) Organization: CSIRO Division of Mathematics and Statistics, Australia Lines: 163 Australian Pattern Recognition Society 2nd CALL FOR PAPERS DICTA-93 2nd Conference on - DIGITAL IMAGING COMPUTING: TECHNIQUES AND APPLICATIONS Location: Macquarie Theatre Macquarie University Sydney Date: 8-10 December 1993. DICTA-93 is the second biennial national conference of the Australian Pattern Recognition Society. This event will provide an opportunity for any persons with an interest in computer vision, digital image processing/analysis and other aspects of pattern recognition to become informed about contemporary developments in the area, to exchange ideas, to establish contacts and to share details of their own work with others. The Following invited speakers will provide specialised presentations: Prof Gabor T. Herman, University of Pennsylvania on Medical Imaging. Prof. R.M. Hodgson, Massey University New Zealand on Computer Vision. Prof. Dominique Juelin, Centre de Morphologie Mathematique, Paris on Mathematical Morphology. Prof. John Richards, Aust. Defence Force Academy, Canberra on Remote Sensing. Dr. Phillip K. Robertson, CSIRO Division of Information Technology, Canberra on Interactive Visualisation. The conference will concentrate on (but is not limited to) the following areas of image processing:- * Computer Vision and Object Recognition * Motion Analysis * Morphology * Medical Imaging * Fuzzy logic and Neural Networks * Image Coding * Machine Vision and Robotics * Enhancement and Restoration * Enhancement and Restoration * Visualisation * Industrial Applications * Software and Hardware Tools Papers are sought for presentation at the conference and publication in the conference proceedings. Submission for peer	82	U I EXAMPLE Labeled UNLABEL

Anchor Name	Cov	% Neg	% Pos	Viz	Model	Delete
ai	0.47% (54)	0.0% (0)	100.% (54)	✓	✓	⊗
Weight: artificial intelligence,neural network,computer vision						
conf ex	50.6% (5736)	99.3% (5700)	0.62% (36)	✓	✓	⊗
conf	1.36% (154)	96.1% (148)	3.89% (6)	✓	✓	⊗
religion distract	17.9% (2026)	99.0% (2006)	0.98% (20)	✓	✓	⊗

Figure 1: An example of a rendered dashboard from ICAT. Throughout the dashboard, blue points and text indicate uninteresting, orange indicate interesting. (a) The AnchorViz ring. (b) The data manager, explorer, and labeling tool. (c) The anchor list/feature editing section.

The dashboard also includes a data manager, shown in Figure 1(b), with tables containing various subsets of the data to make it easier to scroll through and explore the text. The data tabs have five filters that can be applied to the data. These include showing only the current sample of points in the AnchorViz display, showing all points that have been labeled, showing all points the model has predicted are interesting, or showing the result of the user lasso-selecting arbitrary points in the visualization. The tables also include a set of actions per row, allowing the user to apply labels to the points, create example anchors (by default adding a new `TFIDFAnchor` with the chosen row as the similarity target), and add them to the current sample if they are not already included.

Below the AnchorViz ring is the anchor list, shown in Figure 1(c). The anchor list contains all of the current anchors, statistics about the number of points they cover and their classification breakdown, and the associated controls for modifying their parameters. Every anchor type can have a customized set of controls to display when a corresponding anchor row is expanded in the anchor list. This can consist of any combination of ipyvuetify (Widgetti, 2019) elements stored in the `.widget` instance of the anchor. An anchor type is effectively a wrapper around a function that computes a feature value, and ICAT can support dynamically adding new anchor types to the interface for any class inheriting from `icat.anchors.Anchor`.

As discussed earlier, an important concept for a tool that trains a model on the fly based on user interaction is to respond and train quickly (Fails & Olsen, 2003). *Interactive featuring* means that a single user action could change the feature values for the entire training dataset, which requires retraining the underlying model from scratch. Since all features and labels are user provided, the overall volume and necessary complexity is relatively low, and by default ICAT uses scikit-learn's (Pedregosa et al., 2011) logistic regression algorithm. This results in a training time of a few seconds on most modern laptop hardware with datasets smaller than 50,000 entries when using the default anchor types that come with ICAT.

Acknowledgments

The authors would like to acknowledge the US Department of Energy, National Nuclear Security Administration's Office of Defense Nuclear Nonproliferation Research and Development (NA-22) for supporting this work.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

Amershi, S., Cakmak, M., Knox, W. B., & Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, *35*(4, 4), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

Cakmak, M., Chao, C., & Thomaz, A. L. (2010). Designing Interactions for Robot Active Learners. *IEEE Transactions on Autonomous Mental Development*, *2*(2), 108–118. https://doi.org/10.1109/TAMD.2010.2051030

Dudley, J. J., & Kristensson, P. O. (2018). A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems*, *8*(2), 8:1–8:37. https://doi.org/10.1145/3185517

Fails, J. A., & Olsen, D. R. (2003). Interactive machine learning. *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 39–45. https://doi.org/10.1145/604045.604056

Holoviz. (2018). *Panel: The powerful data exploration & web app framework for python*. https://panel.holoviz.org/. https://doi.org/10.5281/zenodo.3706648

Lai, V., Chen, C., Smith-Renner, A., Liao, Q. V., & Tan, C. (2023). Towards a Science of Human-AI Decision Making: An Overview of Design Space in Empirical Human-Subject Studies. *2023 ACM Conference on Fairness, Accountability, and Transparency*, 1369–1385. https://doi.org/10.1145/3593013.3594087

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.48550/arXiv.1201.0490

Simard, P. Y., Amershi, S., Chickering, D. M., Pelton, A. E., Ghorashi, S., Meek, C., Ramos, G., Suh, J., Verwey, J., Wang, M., & Wernsing, J. (2017). *Machine Teaching: A New Paradigm for Building Machine Learning Systems*. https://doi.org/10.48550/arXiv.1707.06742

Suh, J., Ghorashi, S., Ramos, G., Chen, N.-C., Drucker, S., Verwey, J., & Simard, P. (2019). AnchorViz: Facilitating Semantic Data Exploration and Concept Discovery for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems*, *10*(1), 7:1–7:38. https://doi.org/10.1145/3241379

Widgetti. (2019). *Ipyvuetify: Jupyter widgets based on vuetify UI components*. https://github.com/widgetti/ipyvuetify.