Scaling Up Qualitative Data Analysis with Interfaces Powered by Interpretable Machine Learning

Elena Glassman
MIT CSAIL
Cambridge, MA, USA
elg@mit.edu

Been Kim
MIT CSAIL
Cambridge, MA, USA
beenkim@csail.mit.edu

Julie Shah
MIT CSAIL
Cambridge, MA, USA
julie.a.shah@csail.mit.edu

Abstract
When a researcher has more data than they can ever possibly review and code, qualitative methods become challenging to use. Consider a researcher who wants to iteratively build up a set of labels (“codes”) that capture recurring themes in free-response data by reading and labeling portions of text. They can only label as many portions of text as they can read. If the dataset has thousands of free responses, the researcher can never read them all. Machine learning algorithms can train on the researcher’s initial labels and propagate those labels to unseen free responses. To help researchers trust these propagated labels, the machine learning algorithm’s decisions should be interpretable, i.e., understood by the researcher. We believe qualitative researchers working with these large datasets would significantly benefit from data exploration and coding interfaces powered by interpretable machine learning algorithms.

Author Keywords
qualitative methods; user interface design; big data; interpretable machine learning

ACM Classification Keywords
H.5.m [Information interfaces and presentation]: Miscellaneous
Introduction
Unlike quantitative methods, qualitative methods do not scale up easily to datasets that are so large, the researcher cannot even look at all the data. These datasets could contain millions of tweets (Twitter), discussion forum posts by thousands of students (Massive Open Online Course or MOOC), thousands of email and chat messages (Google), or beyond text, e.g., hundreds of small programs generated by students of programming (large university courses).

While previous generations of Computer Assisted Qualitative Data Analysis (CAQDAS) [10, 1] software supported the traditional toolkit of qualitative researchers, i.e., sorting, searching, and annotating, the newest generation of tools is adding features powered by machine learning. Unsupervised machine learning methods can suggest patterns in the data that represent potentially real phenomena that the researcher might never stumble upon while reviewing the small random fraction of data they can read. Supervised machine learning algorithms, once trained with human-provided labels, can approximate the researcher’s judgment on all the items in a large dataset.

However, many machine learning methods are not optimized for human interpretability. Interpretable machine learning methods communicate the mathematical or logical “Why?” behind their judgments to humans and may even let humans directly interact with the algorithm to inject their knowledge or better understand the system output [7]. To be helpful to qualitative researchers, these machine learning methods must be able to communicate with the researcher, leverage their judgment and expertise, and share useful information or patterns from the data [7]. Without interpretability, the researcher may have a hard time trusting [15], debugging [9], or building theory on top of the machine learning algorithm's judgments. Software that incorporates interpretable machine learning methods is key to most effectively scaling qualitative analysis up to large datasets.

Related Work
Software for Qualitative Data Analysis
Computer Assisted Qualitative Data Analysis (CAQDAS) [10] refers to the process of using software to assist in qualitative analysis, e.g., exploring datasets and categorizing data in order to compose and test hypotheses. This software lets researchers focus more on data analysis and less on data management, by providing automated support for sifting, sorting, searching, and annotating patterns and idiosyncrasies in large datasets [1, 11].

Currently available CAQDAS software offers features like fast query-based text retrieval, concordance displays to show context around retrieval results, attaching labels to data, measuring inter-rater reliability of those labels, labeling relationships between data, retrieving data based on labels, topic modeling of textual content, grammatical search, and concept maps. More recently developed software, like the Coding Analysis Toolkit [11], provides better interfaces and additional features to established software, like Atlas/ti [12], or tailored features for particular populations, like journalists (e.g., AP’s Overview Project [14]) and humanities scholars (e.g., WordSeer [13]). WordSeer is targeted at humanities scholars who explore themes across smaller collections of larger documents, e.g., Shakespearean plays or slave narratives. DiscoverText [2] helps researchers pre-process and explore large numbers of small, unstructured units of text, e.g., social media posts.

Algorithms for Interpretable Machine Learning
Interpretable machine learning methods use models that humans can understand, or which at least have internal
states that humans can understand, as well as a human-understandable reason for how the model is derived from the data [5, 7]. These models, without modification, can serve as a bi-directional conduit of information between humans and the machine [8]. Interpretable models include but are not limited to sparse linear classifiers, which make predictions based on adding, subtracting, and multiplying a few meaningful numbers [16]; discretization methods, e.g., decision trees and decision lists [3]; and prototype- or case-based classifiers, e.g., nearest-neighbor-based classification [4].

Case Study
OverCode [6] is an example of an analysis pipeline that can be used for qualitative data analysis on thousands of student solutions to beginner Python programming problems. OverCode's backend uses both static and dynamic analysis to cluster similar solutions; during the process, the highly individualized student programs are mapped into a more canonical form that is optimized for human readability. Various interfaces have been built on top of the OverCode analysis pipeline to help teachers do qualitative analysis on hundreds or thousands of student solutions. For example, teachers of large residential or online programming classes could use this qualitative data analysis to, for example, help identify common ways in which students write poor code and uncommon ways in which students write terrific code.

OverCode's backend clusters similar solutions, but can only handle certain kinds of variation within each cluster. Interpretable clustering algorithms that the teacher can trust are one way to help them better review and understand the full distribution of student solutions. The Interactive Bayesian Case Model (iBCM) [7] is an interactive interpretable machine learning method that is well suited to clustering the canonicalized student solutions produced by the OverCode analysis pipeline. It learns a prototypical example of a cluster: a complete, executable student solution. It also has a mechanism by which users (teachers) can express their own internal metrics about which student solutions are close to each other and which features of code represent important distinctions between solutions.

We created a new interface for displaying iBCM-derived clusters of OverCode-canonicalized solutions, and ran a small study with twelve Python programming teachers as subjects. The teachers interacted with the system to cluster student solutions from a massive open online course (MOOC) on introductory Python programming. They were encouraged to explore variations across hundreds of student solutions before designing a grading rubric or composing feedback for students. The teachers' comments included statements like, “I was able to pull out examples that were really representative of a lot of the solutions and then skim through each solution set to find the minor differences.” A few also noted that the system is “useful with large datasets where brute-force would not be practical” [7].

It is, of course, possible to cluster student solutions using machine learning methods that make no special effort to communicate with the humans receiving the clusters. The method would not explain to the teacher why tens or hundreds of student solutions were clustered together. If a teacher perceives a pattern among the random samples they can read from a large cluster of solutions, they may guess at an explanation, which may or may not be right. Regardless, the teacher may not be able to trust the results enough to make grading-related decisions based on a non-interpretable algorithm's clustering output. This is why we believe that interpretable machine learning methods may be critical to scaling up human judgment of datasets with items too complex, nuanced, or plentiful to be read individually.
Summary
We believe that adapting more traditional qualitative research interfaces to represent the results of interpretable machine learning algorithms is key to scaling qualitative research up to the large datasets now available for research. This will require both smart interface design and continued work on interpretable machine learning methods appropriate for qualitative data analysis.

Acknowledgments
We would like to thank the Python programming teachers who worked with our interfaces, and Quanta Computer for funding some of this work.

REFERENCES