Challenges of Applying Machine Learning to Qualitative Coding

Nan-Chen Chen
Rafal Kocielnik
Margaret Drouhard
University of Washington
Seattle, WA 98195, USA
nanchen@uw.edu
rkoc@uw.edu
mdrouhar@uw.edu

Vanessa Peña-Araya
University of Chile
Santiago, Chile
vpena@dcc.uchile.cl

Jina Suh
Keting Cen
Xiangyi Zheng
Cecilia R. Aragon
University of Washington
Seattle, WA 98195, USA
jinasuh@uw.edu
xiangz3@uw.edu
cenkt@uw.edu
aragon@uw.edu

Abstract

Coding is an important part of qualitative analysis in many fields in social science. Most applications of qualitative coding require detailed, line-by-line examination of the data. Such analysis can quickly become very time-consuming even on a moderately sized dataset. Machine learning techniques could potentially extend the principles of qualiitative analysis to the whole dataset given proper guidance from a qualitative coder. Consequently these techniques offer a promising approach to scale up the coding process. A number of profound challenges, however, still exist that hinder the widespeard use of machine learning for that task. In this paper, we identify a set of challenges for applying machine learning in qualitative coding practice and propose a few directions for future research in this area.

Author Keywords

Machine learning; Qualitative coding; Coding; ML; Social Science; Qualitative analysis

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

Qualitative coding, or simply *coding,* is one of the major techniques used in qualitative analysis among social scientists [17]. In general, coding refers to the process of assigning descriptive or inferential labels to chunks of data, which may assist concept or theory development [13, 14, 19]. Coding is usually a very labor-intensive and time-consuming task [17, 24]. It requires that researchers read through their data in detail, find relevant or potential points of interest, and assign labels. As the scale of datasets grows tremendously in the era of big data, performing such a task on the whole dataset is not feasible for social scientists. As a result, scientists can only sample and code a small part of their data. Since a large portion will remain under-explored, researchers may not be able to resolve inconsistencies in their theories and may not even recognize if some analysis is missing or incomplete.

There have been some attempts to facilitate the coding process for large datasets through fully-automatic or semi-automatic methods. For example, Yan et al. proposed using natural language processing (NLP) and machine learning (ML) to generate initial codes and asking humans to correct the codes. Other work also requires NLP to derive potential codes and/or learn models [5, 6, 9, 12, 19, 24]. While low accuracy has been considered the primary limitation of such automated approaches, we highlight a few other concerns in applying ML to qualitative coding. In this paper, we will provide some background on qualitative coding and illustrate particular conflicts between coding and ML that have hampered the progress in applying ML tools in the qualitative coding domain. We also provide some directions for future research to accelerate the development of applying ML to coding.

Background

The need for qualitative coding

Coding is a common approach to qualitative analysis of data in social sciences [17]. It is a process of arranging qualitative data in a systematic order by segregating, grouping and linking it in order to facilitate formulation of meaning and explanation. Such analysis is often used to search for patterns in the data by organizing and grouping similarly coded data into categories based on commonly shared characteristics [16]. Coding is a necessary because the data in qualitative analysis has no intrinsic organizational structure that explains the phenomenon under study. Researchers must, therefore, create structure and impose it on the data to determine how best to organize the information and facilitate its interpretation for their purposes [11].

Grounded Theory

Grounded theory is one of the most well-known approaches to deal with code organization and theory development. This analysis process involves a number of steps, and they are not carried out sequentially, since insights or realizations of analytics connections can happen any time during the research process [3]. Practitioners of grounded theory often start by reading all of the data repeatedly to achieve immersion [18]. In the next step, the data is coded line-by-line, which is meant to prompt closer study of the data. The initial unrestricted coding is termed *open coding* and is meant to preclude biasing the outcome of coding with preconceived constructs. At this stage such codes are entirely provisional and prone to change. As the analysis progresses, some core variables related to the research topic may be discovered, and the initial codes related to these variables may be applied directly in another round of coding. This form of coding is referred to as *focused* or *closed coding* and it accelerates the analytic pace [18]. To come up with codes used in focused coding, some researchers also use a method called *axial coding*, which is a way to consider the context and relations of the open codes and link them together to generate more meaningful categories [8]. These emergent categories are used to organize and group codes into meaningful clusters [4]. As there are many variations in how to conduct grounded theory approaches [15], individual researchers may use the method differently. Sometimes, the output from the analysis may not be a complete theory, but a *memo* that synthesizes the coded data and captures key theoretical ideas, which can be produced in any stage of coding.

Challenges in Adapting Machine Learning Approaches in Qualitative Coding

Although machine learning has thrived in the past decades, its application in qualitative analysis is still very limited. One common reason for the limited application is that people who use qualitative methods usually do not have background in ML. Thus, due to the complexity of selecting features, building models, and tuning parameters, it may be difficult for them to construct acceptable models. On the other hand, an ML expert might be able to take the codes that a social scientist has applied to part of a dataset in order to train a classifier to label the whole dataset. However, since very few ML experts have background in social science, they do not have contextual information to engineer good features and to prevent issues like over-fitting. For instance, social scientists are usually interested in sophisticated social phenomena, and thus their codes may indicate more complex concepts underlying the data. These concepts may be hard to capture by commonly used decontextualized features such as counts, word use, or even semantic features, and it will be difficult for an ML expert with no social science background to come up with a way to describe those concepts.

Although limited social science understanding of ML experts is a big challenge in applying ML to coding, we also want to identify other inherent conflicts that discourage social scientists from adopting ML methods in their analytical processes. Some of them are due to inherent differences between optimizing ML models and qualitative coding. For example, to build a good classifier, we usually need predefined categories and a large amount of corresponding labeled data (in supervised learning), or the distributions of the datasets must have some distinct separation (in unsupervised learning). However, neither of these is the case in coding. As coding requires heavy manual efforts, it is hard to label sufficient data for strong machine learning results: at the stage of the open coding, scientists do not have pre-defined categories, but gradually create them through closely reading the text. Even in closed coding, where the categories have been decided, the definition of each category may still evolve and be adjusted as more of the dataset is covered. Even though social scientists may want to label as many codes as they can, their ultimate goal is not to build a machine learnable model, but rather to discover some patterns in their datasets or to answer their research questions. It would be unreasonable to demand that they expend huge efforts trying to improve an ML model. Instead, they may prefer to save the effort for reading and coding more in their natural qualitative analysis workflow.

In addition, machine learning usually performs better on categories that have more instances, but those codes may not be the most interesting to a social scientist. This is related to the conflict between quantitative methods and qualitative methods: In quantitative analysis, data points that appear very few times are usually considered to be noise, but from a qualitative analysis point of view, a code that appears more often may not necessarily be more important than a code that appears only a few times. Since it is very hard for any ML method to capture codes that have sparse instances in the dataset, social scientists may prefer to manually code the raw data rather than spend time trying to tune the models.

Beyond the considerations of effort, even though we can build a model for some codes with high accuracy, the results may not be very informative or reliable from a social scientist point of view because most ML methods work like a black box and do not offer explanation for how decisions are made. In our previous experience, even though we coded the dataset with well-tuned models, it may still be hard to interpret why a data point is marked with a certain code. Without knowing how the results are derived, it may not be easy for social scientists to adopt machine learning in their analytical practices. This is also related to the debate around computer-assisted qualitative data analysis software (CAQDAS): When qualitative analysts began to rely more heavily on computer assistance, some expressed anxiety around how computers might negatively impact qualitative research [23]. A study showed that when people do not have sufficient understanding and experience in the methods they use, they were more easily influenced by the results the software suggested [7]. Similar issues may arise if people start to depend more on machine learning and statistical methods in their qualitative analysis.

Directions for Future Research

In the previous section we illustrated a few challenges of adopting machine learning methods in qualitative coding. In addition to the points we mentioned, much work remains to be done towards applying ML in qualitative coding and extending the coding practices to handle big data. In this section, we suggest some potential directions for future research.

Opening the black box of ML

As we mentioned in the previous section, currently many ML models work only as black boxes. It is hard for users to know what happens inside that box and how decisions are made. Therefore, one research direction is to make machine learning models and results more understandable and interpretable. Some work has been done in this direction. For instance, Kim et al. proposed a generative approach to select and extract human interpretable features [10]. Brooks et al. suggested that using interpretable algorithms are important [2]. Visualization is another approach that is considered to be helpful for interpretability [20, 21]. One example of a visualization designed for interpretability is a self-organizing map to show high dimensional feature space of support-vector machine models [22]. However, research in this direction is still relatively scarce, and further exploration should be made to develop models and tools to make ML more human-understandable.

A more complete understanding of the challenges and potentials in applying ML to coding

In addition to the points we mentioned in the previous section, a more complete understanding of the challenges and potentials for applying machine learning to qualitative coding is required. To improve our understanding, we will need more user studies, such as interviews with social scientists experienced in applying machine learning, or ML experts who have worked with social scientists. User studies on social scientists that have no experience in machine learning can also be helpful. It is possible that machine learning will be most useful in a specific step of the coding process. To surface such a point, we need to better understand the coding challenges that social scientists presently face.

Reimagine the use of ML in coding

Both coming up with an ML model that can code data with high accuracy and convincing social scientists to adopt ML techniques in their coding practice might prove difficult. Hence, it might be that a direction most worthy of investigation is to reimagine the use of ML in coding altogether. One such approach is related to the concept of *machine teaching*, where machines act like students and humans play a role of teachers in a form of interactive learning. This view of the problem has been suggested in work by Amershi et al. [1]. While we agree that interactive machine learning can be powerful, we want to extend the idea to a user's perspective: the power of interactive learning is not only in increasing the accuracy of the ML models, but also in providing a way for the users to reflect on their definitions of concepts, and in suggesting new perspectives from which to examine their data. This view is indeed similar to teaching, where not only the students are learning, but the teachers can learn from the questions students ask as well. Furthermore, as students usually have different issues in understanding a concept, teachers must consider different contexts and constantly refine their ideas. Although it is possible that a machine teaching model approach may not result in a competent model for coding, the effort the teachers (the social scientists) spend will still be valuable. In this case, using ML can be seen as part of the coding process. Specifically, it can serve as a way of pinpointing potential issues in current codes, and we believe more work should be done to find new ways to think about the use of ML in the coding practice. Merely automating the existing human practice might overlook some of the most promising opportunities.

Conclusion

Coding is important for qualitative analysis. In many fields in social science, coding is a critical but often time-consuming task. Thus, machine learning may be a useful approach to scale up the coding practice. In this paper, we suggested a set of challenges in applying machine learning to qualitative coding and a few directions for future research. More work should be done to further accelerate the progress of applying machine learning in qualitative coding.

Acknowledgments

We would like to thank all of the social scientists who have assisted us by providing detailed explanations in their current practice and great insights on the challenges they face.

References

1. Saleema Amershi, Maya Cakmak, W Bradley Knox and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning.

2. Michael Brooks, Katie Kuksenok, Megan K Torkildson, Daniel Perry, John J Robinson, Taylor J Scott, Ona Anicello, Ariana Zukowski, Paul Harris and Cecilia R Aragon. 2013. Statistical affect detection in collaborative chat. In *Proc. of CSCW 2013*, 317-328.
3. Kathy Charmaz. 2014. *Constructing grounded theory*. Sage.
4. Amanda Coffey and Paul Atkinson. 1996. *Making sense of qualitative data: complementary research strategies*. Sage Publications, Inc.
5. Kevin Crowston, Eileen E. Allen and Robert Heckman. 2012. Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, 15, 6: 523-543.
6. Kevin Crowston, Xiaozhong Liu and Eileen E. Allen. 2010. Machine learning and rule-based automated coding of qualitative data. *Proc. Am. Soc. Info. Sci. Tech.*, 47, 1: 1-2.
7. Jeanine C Evers, Christina Silver, Katja Mruck and Bart Peeters. 2011. Introduction to the KWALON experiment: Discussions on qualitative data analysis software by developers and users. In *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*.
8. Tiffany Derville Gallicano. 2013. An example of how to perform open coding, axial coding and selective coding. *The PR Post*. Retrieved from https://prpost.wordpress.com/2013/07/22/an-example-of-how-to-perform-open-coding-axial-coding-and-selective-coding/
9. Justin Grimmer and Brandon M. Stewart. 2013. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, 21, 3: 267-297.
10. Been Kim, Julie A. Shah and Finale Doshi-Velez. 2015. Mind the Gap: A Generative Approach to Interpretable Feature Selection and Extraction. In Proc. of *NIPS 2015*.
11. Margaret D LeCompte. 2000. Analyzing qualitative data. *Theory into practice*, 39, 3: 146-154.
12. Seth C. Lewis, Rodrigo Zamith and Alfred Hermida. 2013. Content Analysis in an Era of Big Data: A Hybrid Approach to Computational and Manual Methods. *Journal of Broadcasting & Electronic Media*, 57, 1: 34-52.
13. Matthew B Miles and A Michael Huberman. 1985. *Qualitative data analysis*. Sage Newbury Park,, CA.
14. William Lawrence Neuman. 2005. *Social research methods: Quantitative and qualitative approaches*. Allyn and Bacon Boston.
15. Nicholas Ralph, Melanie Birks and Ysanne Chapman. 2015. The Methodological Dynamism of Grounded Theory. *International Journal of Qualitative Methods*, 14, 4: 1609406915611576.
16. Johnny Saldana. 2009. An introduction to codes and coding. In *The coding manual for qualitative researchers*, 1-31.
17. Anselm L Strauss. 1987. *Qualitative analysis for social scientists*. Cambridge University Press.
18. Renata Tesch. 2013. *Qualitative research: Analysis types and software*. Routledge.
19. Patrick Tierney. 2012. A qualitative analysis framework using natural language processing and graph theory. *The International Review of Research in Open and Distributed Learning*, 13, 5: 173-189.
20. Vanya Van Belle and Paulo Lisboa. 2013. Research directions in interpretable machine learning models. In *European Symposium on Artificial Neuronal Networks, Computational Intelligence and Machiene Learning*.
21. Alfredo Vellido, JD Martin-Guerroro and P Lisboa. 2012. Making machine learning models interpretable. In *Proc. of the ESANN 2012. Bruges, Belgium*, 163-172.
22. Xiaohong Wang, Sitao Wu, Xiaoru Wang and Qunzhan Li. 2006. SVMV–a novel algorithm for the visualization of SVM classification results. In *Advances in Neural Networks-ISNN 2006*, 968-973.
23. Gregor Wiedemann. 2013. Opening up to Big Data: Computer-Assisted Analysis of Textual Data in Social Sciences. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 14, 2.
24. Jasy Liew Suet Yan, Nancy McCracken, Shichun Zhou and Kevin Crowston. 2014. Optimizing Features in Active Machine Learning for Complex Qualitative Content Analysis. *ACL 2014*: 44.