Semi-Automatic Content Analysis of Qualitative Data

Jasy Liew Suet Yan[1], Nancy McCracken[1] and Kevin Crowston[1,2]

[1] School of Information Studies, Syracuse University

[2] National Science Foundation

Abstract

Qualitative content analysis is commonly used by social scientists to understand the practices of the groups they study, but it is often infeasible to manually code a large text corpus within a reasonable time frame and budget. To address this problem, we are building a software tool to assist social scientists performing content analysis. We present our semi-automatic system that leverages natural language processing (NLP) and machine learning (ML) techniques for initial automatic coding, which human coders then review and correct. Through active learning, these human-verified annotations are subsequently used to train a higher performing model for machine annotation. We discuss design strategies adopted to optimize the system performance.

Keywords: natural language processing, qualitative research methods, semi-automatic content analysis, machine learning, active learning

Citation: Yan, J. L. S., McCracken, N., & Crowston, K. (2014). Semi-Automatic Content Analysis of Qualitative Data. In *iConference 2014 Proceedings* (p. 1128–1132). doi:10.9776/14399

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant No. IIS-1111107. Kevin Crowston is supported by the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contact: jliewsue@syr.edu, njmccrac@syr.edu, crowston@syr.edu

1 Introduction

Social scientists often use content analysis to understand the practices of groups by analyzing texts such as transcripts of interpersonal communication. Content analysis is the process of identifying and labeling conceptually significant features in text, referred to as “coding” (Miles and Huberman, 1994). However, analyzing text is very labor-intensive, as the text must be read and understood by a human. Consequently, important research questions in the qualitative social sciences may rely on insufficient data or may fail to be addressed at all.

Computers offer large-scale processing capabilities to deal with systematic patterns in data. However, computers are still not able to truly understand the more subtle meanings in text, so full automation of qualitative content analysis is not yet possible. Furthermore, many natural language processing (NLP) and machine learning (ML) techniques require training on a large amount of coded data, which is time-consuming to produce.

To make qualitative content analysis more scalable, we propose a semi-automatic system that uses a small set of hand-coded examples created by human coders to build a model that can perform a first pass of coding. Human coders then review and correct machine identified instance of codes. These human-verified machine annotations are then used as additional training examples to improve the performance of the ML model. Using this “active learning” approach, we create a significantly larger pool of training examples in a reduced time frame. This paper presents the framework of our proposed approach, and reports some preliminary findings in our initial efforts to optimize the configuration of ML models to perform automatic coding.

2 Related Work

Many computer-assisted qualitative data analysis software (CAQDAS) tools have been developed to support text analysis, but these are not intended for full automatic coding (Alexa, 1997). Researchers have attempted to automate content analysis by applying NLP and ML technologies to identify linguistic patterns in text. For example, Crowston et al. (2010) manually developed NLP rules to automatically identify codes related to group maintenance behavior in free/libre open source software (FLOSS) teams. Ishita et al. (2010) used ML techniques to automatically classify sections of text within documents on ten human values taken from the Schwartz's Value Inventory. Broadwell et al. (2012) developed language models to classify sociolinguistic behaviors used to infer social roles (e.g., leadership). The accuracy of these approaches on the best performing codes ranges from 60-80%, showing the potential of automatic qualitative content analysis. However, these prior studies have been limited to a particular set of theoretical concepts, limiting their general utility.

3 Approach

Figure 1 shows the three major components in our proposed semi-automatic approach: 1) human annotation, 2) machine annotation, and 3) human correction of machine annotation.

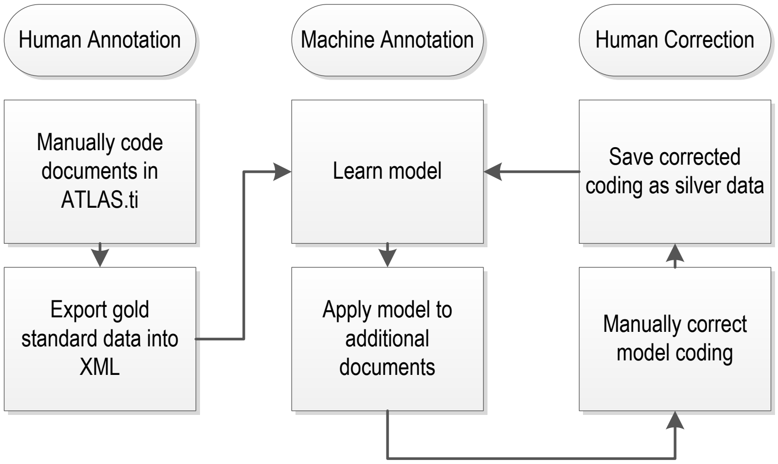
Human Annotation
Machine Annotation
Human Correction
Manually code documents in ATLAS.ti
Learn model
Save corrected coding as silver data
Export gold standard data into XML
Apply model to additional documents
Manually correct model coding

Figure 1: Major system components

Human Annotation: Human coders first manually code a sample of the corpus (in ATLAS.ti, a CAQDAS package) to develop gold standard data for machine annotation. Once manual coding has been finalized, the coded text in ATLAS.ti is exported in an XML format.

Machine Annotation: The gold standard data from the human coders is used to train a support vector machine (SVM) model using pre-selected features and parameters. We approach the machine annotation problem as a text classification task, classifying sentences from the corpus as containing or not containing various codes.

Human Correction: Machine annotations are corrected by human coders. The human-verified annotations are saved as "silver data", and subsequently added to the training set to enhance the performance of the existing model through active learning. This human feedback loop grows the training set gradually, rather than requiring a large initial coding effort.

The model performance is assessed by a combination of recall and precision. Recall measures the ability of the model to find all instances of a code in the corpus, whereas precision measures the percentage of instances returned that are correctly classified. In our system, we emphasize recall, because if recall is high enough, the annotator can depend on the system to find most instances of codes rather than searching the text manually. To achieve high levels of recall, precision will be low, at least initially. Even though human coders will have to review a number of false positives, the system should still save time in large-scale content analysis, as the coders have to read only a subset of the corpus. Adding the human corrected data should improve the precision of the model for future rounds, ideally to the point that the system will produce accurately-coded data without human input, though practically, a final human review may still be necessary.

4 Preliminary Findings

We report preliminary findings from our efforts to optimize the configuration of ML models to perform the first pass of machine annotation. For these tests, we use a gold standard corpus created in a study of leadership behaviors exhibited in emails from a FLOSS development project (Misiolek et al., 2012). This gold standard corpus consisted of 408 email messages. There were a total of 39 codes in the coding scheme. Sentences may be assigned more than one code. Framing the coding as a multi-label classification task, we trained a binary model for each code using SVM with ten-fold cross-validation. These results do not use any “silver data”.

To date, the best model (the one that resulted in the highest recall in model learning) uses only lowercase unigrams as features, with certain specific tokens such as numbers and hyperlinks substituted with more generic tags (e.g., all occurrences of numbers are substituted by <num>). The highest average recall we achieved for all 39 codes is 0.702, meaning that the model is able to detect 70% of positive instances on average from the corpus. On the other hand, average overall precision is only 0.078. Table 1 highlights the top five individual codes with the highest recall. As expected given the high recall, the precision is low.

Code	Gold Frequency	Precision	Recall
Approval	12	0.062	0.95
Commit/Assume Responsibility	17	0.041	0.935
Apology	9	0.026	0.929
Phatics/Salutations	116	0.174	0.896
Inclusive Reference	146	0.336	0.873

Table 1: Top five codes from FLOSS pilot data with the highest recall

Figure 2 shows the distribution of all codes into four quadrants based on the level of recall and precision. Recall and precision values above 0.5 are considered to be high; 0.5 or below, low. A good model would have results in the high/high quadrant, but none of our codes currently reach this level. For the majority of the codes (30 out of 37), system performance falls into the quadrant with high recall and low precision, reflecting our strategy to tune the model for high recall even at the expense of low precision. A model with low recall and high precision might result in a more accurate model but will also miss out many positive instances, which could result in invalid conclusions when the coded data are analyzed. None of the codes fall into this quadrant.

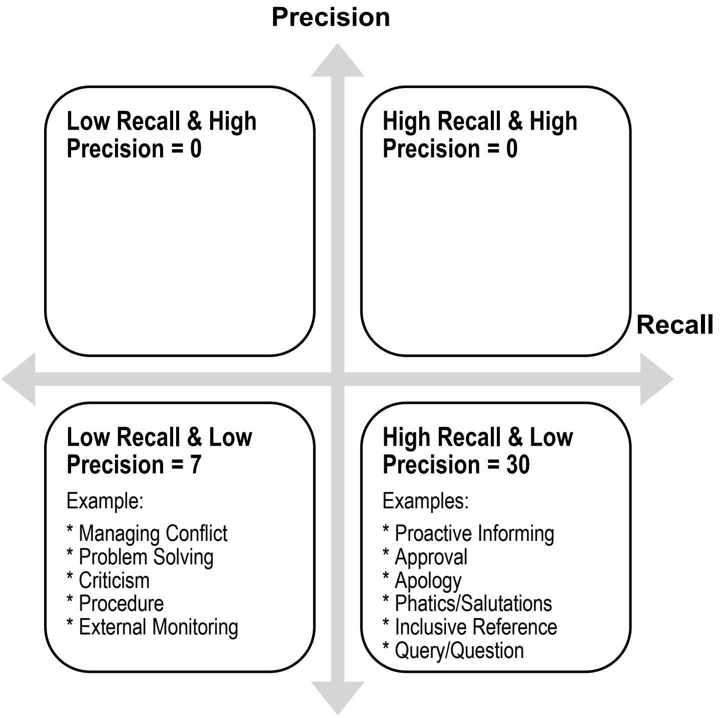

Figure 2: Distribution of individual codes based on initial recall and precision

Finally, for seven codes, the model exhibits both low recall and low precision, an undesirable outcome. Five out of seven codes had fewer than ten examples each in the gold standard corpus, which were too few examples to effectively train a useful model. Providing more instances of these codes in the gold standard data may improve performance. However, two other codes (Problem Solving and Managing Conflict) had more examples in the gold standard corpus but still fell into the same quadrant. Further consideration of these codes reveals that the theoretical concept being captured is actually a complex process rather than a simple behavior. As a result, the coded sentences included considerable variation, which is hard for a model to learn. Indeed, the human coders independently reached the same conclusion, and these two codes have subsequently been removed from the code book.

5 Conclusion and Future Work

We have presented the framework of our semi-automatic approach to content analysis of qualitative data, and explained the two design strategies we have adopted for our system: 1) tuning the ML model performance to emphasize recall rather than precision, and 2) using active learning to continuously train models to yield better results over time. Using this proposed approach, our ultimate goal is to help computers and humans (i.e., social scientists) work closely together to perform large-scale content analysis of qualitative data in a reliable fashion. This paper reports preliminary findings from the implementation of our first design strategy to optimize the configurations of ML models for initial automatic coding.

As part of our future work, we will continue to experiment with different features and model parameters that can further improve the recall of the results for each code of interest. We are currently working on the implementation of our active learning strategy through the creation of silver data being incrementally fed back as training sets for model enhancement. Finally, we hope to encourage more social scientists to help us pilot test this system with other kinds of research data.

6 References

Alexa, M. (1997). Computer-assisted text analysis methodology in the social sciences.

Broadwell, G. A., Stromer-Galley, J., Strzalkowski, T., Shaikh, S., Taylor, S., Liu, T., Boz, U., Elia, A., Jiao, L., & Webb, N. (2013). Modeling sociocultural phenomena in discourse. Natural Language Engineering, 19(02), 213–257.

Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. International Journal of Social Research Methodology, 15(6), 523–543.

Handel, M., & Herbsleb, J. D. (2002). What is chat doing in the workplace? In Proceedings of the 2002 ACM conference on Computer supported cooperative work (pp. 1–10). New York, NY, USA: ACM.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. IEEE Software, 18(2), 16–20.

Ishita, E., Oard, D. W., Fleischmann, K. R., Cheng, A.-S., & Templeton, T. C. (2010). Investigating multi-label classification for human values. Proceedings of the American Society for Information Science and Technology, 47(1), 1–4.

Miles, M. B., & Huberman, A. M. (1994). Qualitative data analysis: An expanded sourcebook. Sage Publications.

Misiolek, N., Crowston, K., & Seymour, J. (2012). Team dynamics in long-standing technology-supported virtual teams. Presented at the Academy of Management Annual Meeting, Organizational Behavior Division, Boston, MA.

7 Table of Figures

8 Table of Tables