Machine Learning and Rule-Based Automated Coding of Qualitative Data

Kevin Crowston, Xiaozhong Liu, Eileen Allen & Robert Heckman
Syracuse University School of Information Studies
Hinds Hall, Syracuse, NY 13244 USA
{crowston,xliu12,eeallen,rheckman}@syr.edu

ABSTRACT

Researchers often employ qualitative research approaches but large volumes of textual data pose considerable challenges to manual coding. In this research, we explore how to implement fully or semi-automatic coding on textual data (specifically, electronic messages) by leveraging Natural Language Processing (NLP). In particular, we compare the performance of human-developed NLP rules to those inferred by machine learning algorithms. The experimental results suggest that NLP with machine learning can be an effective way to assist researchers in coding qualitative data.

Keywords
Natural language processing, machine learning, qualitative data analysis

1. INTRODUCTION

Researchers often apply qualitative research methods to analyze the work practices of groups. For example, researchers might examine transcripts of a group's discussions to understand how it solved some task and the impact of different approaches to that process. Because group artifacts are often textual, they can require considerable manual effort to analyze, as researchers read and reread them to locate evidence to support or refute their theories. This analysis process is referred to as content analysis, more specifically, as coding, as specific passages in the text are tagged as evidence for the various concepts of interest.

In this paper, we discuss the use of natural language processing (NLP) to code qualitative data for social science research. The particular contribution of this paper is to compare preliminary experiments with rule-based and machine-learning-based NLP methods. We present an example study comparing the different techniques: both human coding and NLP coding, the later using both hand-developed and machine-learning-based rule sets. We discuss data and analysis before turning to a comparison of preliminary results. We end with plans for future research.

2. DATA

The example study was an examination of group maintenance behaviours in online groups, that is, behaviours that serve to keep the group together and functioning rather than behaviours directly contributing to the task output (Ridley, 1996). In this short paper, we focus on methodological issues in this study and so do not discuss the substantive research further. The qualitative data we used for this research is typical of research on computer-supported groups, namely the email and discussion forum conversations among free/libre open source software (FLOSS) developers. It is worth noting that such data often exhibit numerous features that challenge traditional NLP algorithms, such as grammar and spelling mistakes, slang, embedded source code fragments and emoticon usage. For this study, we randomly selected 1,469 messages from the developer discussion forums for two FLOSS projects. A random sample of messages was used because the available human coder time was not sufficient to code the entire archive, the problem we hoped to address by using NLP.

Manual coding. Two PhD students trained to code according to a coding scheme derived from the literature. Table 1 shows the detailed constructs explored in this paper (a subset of the full scheme). An iterative process of coding, inspection, discussion and revision was carried out to inductively learn how the indicators of the relevant concepts evidenced themselves in the data, until the coders reached a solid coding scheme. Training continued until the coders reached an inter-rater reliability of 0.80, a typical level expected for human coding. The human coded data were used as the "gold standard" to train and to assess the performance of the NLP coding. These data consisted of the selected messages with short phrases identified and coded that express the various theoretical constructs.

3. AUTOMATIC CODING

The goal of the research was to develop NLP techniques to support qualitative research by automating (to the extent possible) the qualitative coding process. Coding was approached as an information extraction problem, meaning that the NLP software was used to extract from the textual data phrases that are evidence for the theoretical concepts of interest (Cowie & Lehnert, 1996; Appelt, 1999; Cunningham, 1999). In this paper, we compare two methods for extracting the coded text: a rule-based approach and a machine-learning based approach.

Rule-based approach

In the first approach, we developed and applied human-developed NLP rules to extract the coded segments. This approach is knowledge-based, analyzing linguistic phenomena that occur within text using syntactic, semantic and discourse information. For this study, an expert NLP analyst developed rules using an NLP program developed in the authors' research centre (anonymized for review). To develop the rules, the analyst reviewed the codebook to provide a top-down understanding of the constructs, and also reviewed the marked data to gain a bottom-up understanding of how the codes were interpreted and implemented in the text. The rule-writing process was iterative, whereby rules were written to code the most abundant and obvious examples of the coded text, and then progressively refined for coverage and accuracy. Rule-writing was interspersed with testing to assess performance on the training data during the development process. A portion of the coded data (155 messages, or about 10%) was reserved for testing of the completed ruleset. The remainder was used to assess the performance of the ruleset as it was being built.

Some rules, as for *Capitalization*, were primarily based on regular expressions to detect upper case. Other rules, as for *Apology*, focused on specific lexical items—'sorry', 'apologies'—or a lexicon of items. But others, such as the rule for *Agreement*, required the use of the full range of features such as part of speech, token string and syntax.

Indicator	***Definition***
Emoticons	Emphasis using emoticons
Capitalization	Emphasis using capitalization
Punctuation	Emphasis using punctuation
Slang	Use of colloquialisms or slang beyond group-specific jargon
Inclusive pronouns	Incorporating writer and recipient(s)
Complimenting	Complimenting others or message content
Agreement	Showing agreement
Apologies	Apologizing for one's mistakes
Encouraging participation	Encouraging members of the group to participate
Appreciation	Showing appreciation for another person's actions
Hedges/Hesitation	Tactics to diminish force of act; hesitation in disagreement

Table 1: Code book for group maintenance behaviours.

Hmmm.... the "real" one should be at >>> /fire/.lproj /MailControllerWindow.ni* *[Code: Hedges/Hesitation]*

ya but when ur typin in an im u always spell things howevere is da shortest way. *[Code: Slang]*

u guys are great *[Code: Complimenting]*

Figure 1. Examples of coded textual data.

There are two advantages to the rule-based approach. First, the rule-based approach is not necessarily sensitive to the number of examples available, as human experts develop the rules and can apply their expert knowledge. Second, the experts can also adjust the rules in light of the nature of the data, e.g., to compensate for spelling and grammar mistakes. However, the cost of this approach is high, since it requires effort from a trained professional, though that cost may be spread across a large volume of data to be analyzed.

Machine-learning approach

The second approach used machine-learning (ML) algorithms to automatically learn the complex patterns underlying the extraction decisions based on the statistical and semantic features identified in the textual data. Again, a portion (75%) of the human-coded data was used for training and the remainder for testing. Compared with manual rule-writing, the ML process was more automatic. The training data were used to train a classifier using a ML algorithm that inferred rules for extraction using features of the messages.

The ML algorithm used in this experiment was Winnow (Littlestone, 1988), which is a linear classifier that works by updating the weights assigned to the different features. We chose Winnow for three reasons. First, Winnow has been successfully used for information extraction problems, e.g., by Zhang et al. (2002). Second, Winnow is an easy and effective online learning algorithm. The inferred decision rules can be easily updated with further training instances, which could be used to incorporate feedback from human coders in a semi-automated learning process. Finally, it is known that Winnow is an effective algorithm in the presence of irrelevant attributes (Littlestone 1988, Dhagat & Hellerstein 1994), which we expected given the nature of message data.

The performance of the ML depends on correct selection of the features in the text that should be used for the rules to be learned. In these initial experiments, we compared performance using three simple sets of features:

ML (BOG, LOC): Bag-of-words and location only. Only the token strings and the location of a word are employed, for example, one word or two words before or after the target coding result.

ML (BOG, POS, LOC): As above, plus part-of-speech.

ML (BOG, POS, CAP, LOC): As above, plus capitalization, whether the first character of the token is capitalized.

For all tests, a [-3, 3] text window (all six tokens) around the target coding result was used to define the feature space.

Using ML to infer rules can be more cost-effective than the rule-based approached as it does not require the time of an expert to write the rules (which is not to say that expertise is not required at all). However, performance of the ML approach is highly dependent on having a large number of training examples from which to learn and being able to identify a useful semantic feature space on which to learn. Unfortunately, we have only a few examples for many of the codes in this study. Further, grammar, spelling and capitalization mistakes and frequent use of domain-specific proper nouns may make it hard to create a usable feature space. For instance, the irregularity of the examples shown in Figure 1 could pose problems for inferring rules.

4. EXPERIMENTAL SETUP

In this section, we describe the experiment we conducted to compare the performance of the two approaches to NLP as applied to qualitative data coding, and compare both to the human coded data used as the gold standard data.

For the analysis, the message data were pre-processed to convert them into a standard format that would preserve the metadata elements (e.g., sender, date, subject), identify significant features of the data, such as signature lines or quoted messages and prepare the data for processing with our text processing engine, thus encoding the discourse structure for further use. The steps included:

1. Sentence splitting. Because some message writers did not use grammatical punctuation, the performance of the splitter is not as high as on a regular corpus.
2. Tokeniser, to split sentences into simple tokens. Tokens in the data can be quite long, such as unique file names or unix command names.
3. Part-of-speech (POS) tagger. Each token was tagged with its part of speech using sentence level context information.

For the human-written rules experiment, the data were prepared as text with POS tags applied to the lemmatized words (e.g., the element 'do|VBZ' combines a lemma, the lexical item, 'do' and the part of speech 'VBZ', present tense verb. For the ML experiment, the sentences in the corpus with the pre-processing tags and the human applied codes were prepared in a XML format.

5. EXPERIMENTAL RESULTS

The experimental results for both NLP approaches (with the 3 different feature sets for ML) are displayed in Table 2. Two standard information extraction metrics were used to evaluate the automated system, Recall and Precision. Recall measures the proportion of the codes in the gold standard data that was identified and extracted by the system (i.e., coverage). Precision measures the proportion of the automatically extracted data that was coded correctly, as compared to the gold standard data (i.e., accuracy).

It is usually difficult to have high performance on both measures: the more accurate the results, the smaller the coverage of the target data and vice versa. To completely automate coding, it would be necessary to achieve good performance on both measures. In building the rules, the decision was made to optimize the automated system for Recall, with a goal of 80%, under the assumption that it is easier for a human reviewing the system output to remove incorrectly coded data (due to low Precision) than to search entire email logs to find evidence that had not been coded at all (the result of low Recall). The last column shows the size of the training set for the machine learning approach, as it affects performance.

We start by examining the performance of the manually developed ruleset. Examining the codes in more detail, Recall was highest for *Emoticon* and *Inclusive Pronouns*, reflecting the regularity of the realization of these constructs in the text. Recall was lower for codes such as *Slang* or *Appreciation* that show higher variability. The Precision of the results is lower, reflecting our decision to favor Recall over Precision. Nevertheless, Precision is quite good for a number of codes, such as *Emoticon* or *Salutations*, and with

CODE	**Rule-based results**		**ML (BOG, LOC)**		**ML (BOG, POS, LOC)**		**ML (BOG, POS, CAP, LOC)**		**Training Size**
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	
Apologies	67%	67%	0%	0%	0%	0%	50%	100%	5
Complimenting	40%	67%	0%	0%	0%	0%	0%	0%	36
Agreement	60%	80%	60%	23%	0%	0%	73%	31%	104
Capitalization	19%	60%	0%	0%	0%	0%	0%	0%	20
Appreciation	45%	64%	54%	67%	56%	67%	50%	60%	60
Emoticon	81%	91%	48%	58%	22%	56%	38%	53%	144
Salutations	86%	86%	77%	80%	100%	68%	87%	52%	105
Punctuation	22%	71%	65%	48%	72%	46%	63%	45%	268
Slang	69%	67%	50%	4%	64%	8%	50%	7%	384
Inclusive Pronouns	58%	98%	93%	93%	95%	93%	92%	90%	240
Hedges/ Hesitation	69%	74%	47%	43%	62%	45%	58%	48%	1276

Table 2: Experimental results comparing the NLP approaches to the human coded data.

the exception of *Capitalization* and *Punctuation*, all are at usable levels.

Turning to the ML results, we note that the results are extremely poor for codes with very few instances in the training set. However, given a sufficient number of training example, the performance of the ML rules improve, with the conspicuous exception of *Slang*. The ML rules even match the human-created ruleset for a few codes, such as *Inclusive pronouns*.

In order to compare the three feature sets used in the ML approach, we present the comprehensive evaluation results across sets in Table 3. Overall, there does not appear to be much difference. Interestingly, it was not the case that more features always led to better performance. For a number of codes, performance with just the words and location performed best.

ML (BOG, LOC)		**ML (BOG, POS, LOC)**		**ML (BOG, POS, LOC, CAP)**	
Precision	Recall	Precision	Recall	Precision	Recall
56%	41%	59%	42%	61%	44%

Table 3. Results for feature sets for machine learning

6. CONCLUSION

From the experimental results, we make the following initial conclusions:

1. Both rule-based and machine-learning-based automatic coding seems to offer promise for coding qualitative data, even in the face of the low quality of the text.
2. Overall, the rule-based approach performed better than the machine-learning approach, especially for those codes with few training instances. Evidently, the expert can generate high-quality rules from a small number of training instances and make good decisions about the appropriate feature(s) to use for different codes.
3. For the ML approach, the capitalization and part-of-speech features sometimes improve learning performance and sometimes not. However, it is clear that much more work needs to be done to identify good feature sets for this application of ML.

In the future, we will work in two directions. First, we plan to search for better semantic features and examine the use of different machine learning algorithms to improve the machine learning approach. The work presented in this paper is just a first step in that direction. Second, we plan to implement a coding system that will take coded data as input and provide a mechanism whereby the human coders can correct the NLP output. Within this system, we are interested in exploring how the corrected output could be reinput into the ML as a basis for inferring a refined set of rules. Our choice of Winnow was made with this approach in mind. We expect that with more training instances, automatic coding performance can be improved.

In conclusion, analyzing significant volumes of qualitative data currently requires considerable effort from researchers. The experiment presented above, while preliminary, suggests that using rule-based or machine-learning-based automatic coding could assist researchers to code larger amounts of data at a lower cost than entirely manual coding. Furthermore, our results from applying ML, while again preliminary, are also promising, suggesting that these benefits might be obtainable without requiring the efforts of a highly-trained NLP analyst. In practice, human coders would still have to be used to code an initial set of data for training, but from there the trained classifier could be used to infer the code labels for the rest of the data automatically. Coders could then shift their attention to checking the machine-coded data to further improve precision and to the most important and non-automatable job of making sense of the data.

REFERENCES

Appelt, D. (1999). An introduction to information extraction. *Artificial Intelligence Communications*, 12(3): 161–172.

Cowie, J. & Lehnert, W. (1996). Information extraction. *Communications of the ACM*, 39(1): 80–91.

Cunningham, H. (1999). *Information extraction: A user guide (revised version)*. Research Memorandum CS–99–07, Department of Computer Science, University of Sheffield, May.

Dhagat, A., Hellerstein, L. (1994). PAC learning with irrelevant attributes, *Foundations of Computer Science*, Annual IEEE Symposium.

Littlestone, N. (1988). Learning quickly when irrelevant attributes: A new linear-threshold algorithm. *Journal of Machine Learning*, 2, 285-318.

Ridley, M. (1996). *The Origins of Virtue: Human Instincts and the Evolution of Cooperation*. New York: Viking.

Zhang, T., Damerau, F., and Johnson, D. (2002). Text chunking based on a generalization of Winnow. *Journal of Machine Learning Research*, 2:615–637, March.