Machine Learning and Rule-Based Automated Coding of Qualitative Data

Kevin Crowston
School of Information Studies
Syracuse University,
Syracuse, NY 13244, USA
crowston@syr.edu

Xiaozhong Liu
School of Information Studies
Syracuse University,
Syracuse, NY 13244, USA
xliu12@syr.edu

Eileen E. Allen
School of Information Studies
Syracuse University,
Syracuse, NY 13244, USA
eeallen@syr.edu

ABSTRACT

Large volumes of textual data pose considerable challenges for manual qualitative analysis. We explore semi-automatic coding of textual data by leveraging Natural Language Processing (NLP). We compare the performance of human-developed NLP rules to those inferred by machine learning (ML) algorithms. The results suggest that NLP with ML may be useful to support researchers coding qualitative data.

Keywords

Natural language processing, machine learning, qualitative data analysis

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. Because group artifacts are typically textual, they can require considerable manual effort to analyze, as researchers read and reread them to locate evidence to support or refute their theories, tagging specific passages in the text as evidence for the various concepts of interest. This analysis process is referred to as content analysis, more specifically, as coding. We discuss the use of natural language processing (NLP) technology for coding qualitative data for social science research. The particular contribution of this poster is to compare preliminary experiments with rule-based and machine-learning-based NLP methods for this application.

DATA

We use as an example a study of group maintenance behaviours in online groups, that is, behaviours that serve to keep the group together and functioning rather than directly contributing to the task output (Ridley, 1996). The qualitative data we used for this research are 1,469 randomly selected messages from the developer discussion forums for two free/libre open source software (FLOSS) projects among developers.

ASIST 2010, October 22–27, 2010, Pittsburgh, PA, USA.

Code book development and manual coding. Two PhD students trained to code according to a coding scheme derived from the literature. 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 an inter-rater reliability of 0.80, a level expected for human coding.

AUTOMATIC CODING

Our goal was to develop NLP techniques to automate (to the extent possible) the qualitative coding process. Coding was approached as an information extraction problem: the NLP software extracts from the textual data phrases providing evidence for the theoretical concepts of interest (Cowie & Lehnert, 1996; Appelt, 1999; Cunningham, 1999). In this poster, we compare two methods for developing rules for extracting coded text: a manual approach and a machine-learning (ML) based approach.

Manual Rule Development Approach

In the first approach, an expert NLP analyst developed NLP rules to extract the coded segments. To develop the rules, the analyst reviewed the codebook and manually-coded data to develop an understanding of how the codes were interpreted and implemented in the text. This approach is knowledge-based, analyzing linguistic phenomena that occur within text using syntactic, semantic and discourse information. 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 linguistic features such as part of speech, token string and syntax, which are beyond the capability of currently-used lexicon-based analysis systems.

Machine-Learning Approach

The second approach used a ML algorithm (Winnow, Littlestone, 1988) to learn the complex patterns underlying extraction decisions based on the statistical and semantic features in the textual data. Dönmez et al (2005) report on a similar use of ML. Using machine learning 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 machine-learning 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 of some codes and have just begun to explore the possible feature space.

In these initial experiments, a portion (75%) of the human-coded data was used for training and the remainder for testing. For all tests, a [-3, 3] text window (all six tokens) was used to define the feature space. We compared performance using three sets of features:

1. ML (BOG, LOC): Bag-of-words and location only.
2. ML (BOG, POS, LOC): As above, plus part-of-speech.
3. ML (BOG, POS, CAP, LOC): As above, plus capitalization (whether the first character of a token is capitalized).

EXPERIMENTAL RESULTS

The experimental results for both NLP approaches are displayed in Table 1. For the manually developed ruleset. Recall was highest for the codes *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 a decision to favor Recall over Precision. Nevertheless, Precision is quite good for a number of codes, such as *Emoticon* or *Salutations*, with the exception of *Capitalization* and *Punctuation* (these were affected by the inclusion of source code in the messages that was not coded by the human coders).

For the machine-learning results, the results are 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*. While the manually-developed ruleset did perform better overall, the performance of the ML rules matched the human-created ruleset for a few codes, such as *Inclusive pronouns*. Interestingly, there did not appear to be much difference between the feature sets: more features did not always lead to better performance. For a number of codes, performance with just the simple linguistic features performed best.

CONCLUSION

From the experimental results, we conclude that both rule-based and machine-learning-based automatic coding seems to offer promise for coding qualitative data. However, our results highlight the impediments to applying either approach: the need for either an NLP expert to manually develop rules or a large number of examples from which to infer rules.

To address these problems, we plan to work in two directions. First, we will search for better semantic features and examine the use of different machine learning algorithms to improve the ML performance. The work presented in this paper is just a first step in this direction.

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

Second, we plan to implement a system that will take coded data as input, infer and run a set of rules, and output a coded data set. The system will provide a mechanism for the human coders to correct the NLP output. To address the need for a large training sample of coded data, we will to explore how the corrected output could be reinput to the ML as a basis for inferring a refined set of rules. Such an approach may reduce the amount of initial human coding needed, enabling more widespread application of NLP to the problem of qualitative text analysis.

ACKNOWLEDGEMENT

This research was partially supported by NSF Grants 04–14468, 05-27457 and 07–08437.

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.

Dönmez, P., Rosé, C., Stegmann, K., Weinberger, A., & Fischer, F. (2005). Supporting CSCL with automatic corpus analysis technology. In *Proceedings of the Conference on Computer Support for Collaborative Learning (CSCL).*

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.