Supporting Efficient and Reliable Content Analysis using Automatic Text Processing Technology

Gahgene Gweon[1],Carolyn Penstein Rosé[1], Joerg Wittwer[2], Matthias Nueckles[2]

[1] Human-Computer Interaction Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213
{ggweon, cprose}@cs.cmu.edu
[2] Universitaet Freiburg, Institut Fuer Psychologie,
Engelbergerstr. 41, 79085 Freiburg, Germany
{wittwer,nueckles}@psychologie.uni-freiburg.de

Abstract. Text categorization technology can be used to streamline the process of content analysis of corpus data. However, while recent results for automatic corpus analysis show great promise, tools that are currently being used for HCI research and practice do not make use of it. Here, we empirically evaluate trade-offs between semi automatic and hand labeling of data in terms of speed, validity, and reliability of coding in order to assess the usefulness of incorporating this technology into HCI tools.

1. Introduction and Background

A wide range of behavioral researchers in the field of Human Computer Interaction, such as social psychologists studying communication in Computer Supported Cooperative Work settings, painstakingly code by hand and analyze large quantities of natural language corpus data as an important part of their research. Similarly, research in Computer Supported Collaborative Learning often depends upon quantitative process analyses through multi-dimensional coding schemes such as those described in [3]. Another instance, more relevant for HCI practitioners than researchers, is labeling of Contextual Inquiry data in order to mark instances relevant for different types of models or instances of standard types of critical incidents.

In this paper we explore issues related to the use of technology for supporting such coding by making automatic predictions about those codes. Despite the availability of such technology, none of the tools that are commonly used in the HCI community make use of it. For example, specialized software for supporting text, audio, and video data analysis such as HyperResearch, MacShapa, and Nvivo offer well designed interfaces for hand annotation and cataloguing of codes. Yet, none of them go as far as to support automatic assignment of codes. In this paper, we investigate the possibility of incorporating automatic classification of text into data analysis applications widely used in HCI research and practice. Since automatic coding technology is not perfect, and the data analyzed in HCI research requires subjective judgments, before such technology can safely be accepted into common practice, it is

essential to measure precisely how the accuracy of the classification affects user performance.

2. Technical Approach

Automatic text classification technology for supporting automatic analysis of corpus data shows great potential benefit to HCI researchers and practitioners [1]. Here we offer a brief introduction to this technology. Applying a categorical coding scheme can be thought of as a text classification problem where a computer decides which code to assign to a text based on a model built from examining "training examples" that were coded by hand. Some machine learning techniques that have successfully been applied to this problem are regression models, nearest neighbor classifiers [7], decision trees [4], and Bayesian classifiers [2]. Success at automatic coding can be evaluated by comparing predicted codes to those in a Gold Standard corpus, which is a corpus that has been annotated with a coding scheme, and the coding has been verified to be reliable. For example, in comparison with a hand-coded Gold Standard of a 1200 sentence test corpus coded with the 7 dimensional coding scheme used for the analysis in [5], it is possible with state-of-the-art technology to achieve an acceptable level of agreement (Cohen's $K > .7$) along 6/7 of the dimensions between automatically generated codes and hand coded codes when we commit only to the portion of the corpus where the predictor has the highest certainty [1]. Along 5 of those dimensions, the percentage of the corpus where the predictor was confident enough to commit a code was at least 88% of the corpus. Thus, fully automatic coding is a viable option for the greater portion of the corpus, but not the whole thing.

3. Formal Study

When fully automatic coding is not a viable choice, there are two primary options: (a) Limit automatic predictions to the subset of the data where the predictor can confidently assign a code with high reliability; or (b) Have the system make predictions about everything, and the analyst checks and corrects the codes for dimensions where the reliability is not at an acceptable level. The purpose of our study is to measure the impact of automatic predictions in terms of speed, validity, and reliability of human judgment when the predictions are not accurate enough for fully-autmatic coding. The coding scheme used in the study presented in this paper was established in a study by Wittwer, Nückles and Renkl who examine computer-mediated communication between experts and laypersons in the context of asynchronous help-desk support [6]. 20 participants, students and staff at nearby universities, were randomly assigned to 2 conditions. In the control condition, participants worked with the coding interface with no predicted code (Hand-Code). In the experimental condition, participants worked with the minimally adaptive coding interface that displays predicted codes in such a way that 50% of the sentences were randomly selected to agree with the Gold Standard, and the rest were randomly

assigned (Auto-Code-And-Correct). This proportion of correct versus incorrect predictions was selected based on timing results from a small pilot study conducted before the study reported here. In a small pilot study with 3 expert analysts coding 141 sentences each, we measured the average time required to code a sentence depending upon the selected strategy (Hand-Code vs. AutoCode-And-Correct) and (in the AutoCode-And-Correct case) whether the predicted code is correct. On average the analysts spent 11 seconds/ sentence when they agreed with the prediction. They spent on average 21.5 seconds/ sentences where they did not agree with it. And they spent 17 seconds on average per sentence when no prediction was offered. Thus, the advantage of the automatic predictions in terms of coding time may depend upon the % of time that the coder agrees with the automatic codes.

Procedure. Participants first spent 20 minutes reading the 6 page coding manual. They then spent 20 minutes working through 28 training exercise sentences using the coding manual. They were instructed to think aloud about their decision making process. They then received coaching from an experimenter to help them understand the intent behind the codes and compared their answers with a Gold Standard set of codes. After a 5 minute break, they spent up to 90 minutes coding 76 sentences.

Interface. Participants coded sentences using a menu-based interface. For the standard coding interface, the sentences were arranged in a vertical list. Next to each sentence was a menu containing the complete list of codes. No code was selected as a default. In contrast, a minimally adaptive version was used in the experimental condition. The only difference between this and the standard version was that a predicted code was selected by default for each sentence, which appeared as the initial element of the menu list and was always visible to the analyst.

4. Results and Recommendations

First we evaluated the reliability of coding of each of the two conditions. Average pairwise Kappa measures of agreement were significantly higher in the experimental condition ($p < .05$). Mean pairwise Kappa(K) in the control condition was .39, whereas it was .48 in the experimental condition. As a measure of the best we could do with novice analysts and 50% correct predicted codes, we also analyzed the pairwise K measures of the 3 participants in each condition whose judgments were the most similar to each other. With this carefully chosen subset of each population, we achieved an average pairwise K of .54 in the control condition and .71 in the experimental condition. This difference was significant ($p < .01$). The average agreement between these analysts' codes from the experimental condition and the Gold Standard was also high, an average K of .70. Thus, the analysts who agreed most with each other also produced valid codes in the sense that they agreed well with the Gold Standard. Next, evaluating the validity of coding more stringently, we found analysts in the experimental condition were significantly more likely to agree with the prediction when it was correct (74% of the time) than when it was incorrect (16% of the time). This difference was significant using a binary logistic regression with 760 data points, one for each sentence coded in the experimental condition

(p<.001). Average K agreement with the gold standard across the entire population was marginally higher in the experimental condition than in the control condition (p=.1). Average agreement in the unsupported condition was a K measure of .48. In the experimental condition, average agreement with the gold standard was a K measure of .56. The raw percent agreement with the gold standard was significantly higher in the experimental condition than in the control condition ($p < .001$). Thus, we conclude that analysts were not harmfully biased by incorrect codes. Coding time did not differ significantly between control (67min 36sec) and experimental (66min 10sec) conditions, thus providing some confirmation of the estimate that 50% correct predictions is a reasonable break even point for coding speed.

In this paper we have described some investigations towards developing a tool for providing automatic coding support for human corpus analysts. Our evaluation demonstrates that a savings of time with AutoCode-And-Correct over Hand-Code only occurs when percent agreement between predicted codes and the analyst's selected codes is greater than 50%. At that percent agreement level, we demonstrate that the AutoCode-And-Correct option leads to an increase in reliability while maintaining validity of human judgments by novice coders. Making use of predictions to boost reliability with less skilled, and thus cheaper, analysts may be an advantage in terms of research costs as long as the coding scheme itself is properly evaluated for reliability first. Since 50% prediction is feasible with technology available in the computational linguistics community, the next step would then be to integrate technology with tools used in HCI community for easy, reliable and faster coding.

This work was supported by NSF SGER REC-0411483.

References

1 Donmez, P., Rosé, C. P., Stegmann, K., Weinberger, A., and Fischer, F. Supporting CSCL with Automatic Corpus Analysis Technology, to appear in the Proceedings of Computer Supported Collaborative Learning'05, forthcoming.

2. Dumais, S., Platt, J., Heckerman, D. and Sahami, M. *Inductive Learning Algorithms and Representations for Text Categorization*, Technical Report, Microsoft Research, 1998.

3. Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction*, 12, 213-232, 2002.

4. Lewis, D., Ringuette, R. A Comparison of teo learning algorithms for text classification, In *3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 81-93, 1994.

5. Weinberger, A. Scripts for Computer-Supported Collaborative Learning Effects… 2003, From http://edoc.ub.uni-muenchen.de/archive/00001120/01/Weinberger_Armin.pdf

6. Wittwer, J., Nückles, M., Renkl, A. Can experts benefit from information about a layperson's knowledge for giving adaptive explanations?. In K. Forbus, D. Gentner, T. Regier (Eds.), *Proc. 26th Annual Conf of the Cognitive Science Society*, 2004. 1464-1469.

7. Yang, Y. and Pedersen, J. Feature selection in statistical learning of text categorization, In *the 14th Int. Conf. on Machine Learning*, pp 412-420, 1997.