Survey Research Methods (2016)
Vol. 10, No. 2, pp. 143-152
doi:10.18148/srm/2016.v10i2.6213

ISSN 1864-3361
http://www.surveymethods.org

Semi-automated categorization of open-ended questions

Matthias Schonlau
University of Waterloo
Ontario, Canada

Mick P. Couper
University of Michigan
Ann Arbor, U.S.A

Text data from open-ended questions in surveys are difficult to analyze and are frequently ignored. Yet open-ended questions are important because they do not constrain respondents' answer choices. Where open-ended questions are necessary, sometimes multiple human coders hand-code answers into one of several categories. At the same time, computer scientists have made impressive advances in text mining that may allow automation of such coding. Automated algorithms do not achieve an overall accuracy high enough to entirely replace humans. We categorize easy-to-categorize text answers of open-ended questions automatically using text mining and multinomial boosting, and hard-to-categorize text answers manually. Expected accuracies guide the choice of the threshold delineating between "easy" and "hard" to code text answers. This approach is illustrated with two examples from open-ended questions related to respondents' advice to a patient in a hypothetical dilemma, and a follow-up probe related to respondents' perception of disclosure/privacy risk. Targeting 80% accuracy, we found that 47%-58% of the data could be categorized automatically in research surveys.

Keywords: multinomial boosting, gradient boosting, qualitative data, coding, text mining

1 Introduction

Open-ended questions are often manually coded into different categories. Manual categorization is time consuming and expensive (Geer, 1991), and does not scale well in large surveys. Therefore text data from open-ended questions are often ignored, or sometimes individual text answers are used for anecdotes or qualitative analyses. This paper primarily addresses categorizing narrative text answers. The two examples we use consist of narrative responses to an open-ended question in a Web survey.

Fully automated text mining for narrative open-ended questions is generally not as accurate as manual categorization, which is a problem for researchers who value accuracy over low cost and a fast turnaround time. We focus on categorization with high accuracy rather than full automation. Briefly, we turn text into numerical variables using the ngram approach used in text mining (e. g. Joachims, 2002; Schonlau & Guenther, 2016). Some answers are categorized manually and serve as training data. We next apply multinomial boosting, a statistical learning[1] algorithm, on the training data and compute the probability of correctly categorizing each answer in the test data. We categorize text answers with a high probability of correct classification automatically and the remainder manually. For the subset of test data that is categorized automatically the expected accuracy can be computed. In summary, automated categorizations are used where possible and manual categorization where necessary.

We next give some background on text mining and boosting (Sections 2 and 3) and then describe the proposed approach (Section 4). Our approach relies on approximately unbiased probability estimates of categorization. We use a simulation to show which boosting parameters yield approximately unbiased estimates of the probability of categorization (Appendix B). The proposed approach is illustrated with two examples (Sections 5 and 6). Our approach can achieve substantial time savings (Section 7). We conclude with a discussion (Section 8).

2 Text Mining

Categorizing texts using text mining consists of two steps: In the first step, text is encoded into a set of numeric variables such that in a second step statistical learning algorithms can be employed (Witten, Frank, & Hall, 2011). Briefly, each word ("unigram") is encoded into a separate indicator variable indicating whether or not a text answer contains the word. Several modifications are made including stemming (reducing inflected words to their root form), discarding of common words ("stop words") such as "the" and "and", and discarding words that appear too infrequently. Variables for

Correspondance author: Matthias Schonlau, University of Waterloo, 200 University Ave W, Dep. of Statistics and Act. Sci., Bldg. M3, Waterloo, Ontario, Canada N2L 3G1 (email: schonlau@uwaterloo.ca)

[1]The terms "statistical learning" and "machine learning" are synonyms with one term being used more frequently in the statistical sciences and the other more frequently in computer science.

sequences of two words, "bigrams," are also often created. This approach to text mining is called "set of words" (Hotho, Nürnberger, & Paaß, 2005) because it uses only the presence or sometimes also the frequency of a word, not the order in which it appears in the text.

Encoding text into indicators of unigrams and bigrams is illustrated in Table 1 with three hypothetical texts. Presence/absence of each word is encoded as an indicator variable. The stop word "the" is omitted; "eats" is reduced to its stem "eat". For space reasons only three of the bigrams are shown: "cat_eat", "dog_eat", and "mouse_eat". Including bigrams partially recovers word order and allows distinguishing between "The cat eats the mouse" and "The mouse eats the cat".

This approach to text mining has been implemented in the Stata package *ngram* (Schonlau & Guenther, 2016), in the R programming language (Meyer, Hornik, & Feinerer, 2008), in free-for-non-commercial use software LightSIDE (Mayfield & Penstein Rose, 2012) (English only), as part of the PERL programming language, and elsewhere.

3 Prediction with boosting

After encoding the text into numerical variables, some type of regression is employed to relate the outcome, the multinomial variable that assigns the text to a category, to the x-variables (unigrams, bigrams, and others). Multinomial linear regression does not work well for prediction in text mining for two reasons: a) variables are highly collinear and b) text mining creates thousands of variables; potentially more variables than observations. Variables are highly collinear because text answers typically only contain up to a few dozen words. For any given text, most unigrams and bigrams are zero.

For prediction, statistical learning algorithms are used. A popular statistical learning algorithm is gradient boosting (Friedman, Hastie, & Tibshirani, 2000). We choose this algorithm, because it computes probabilities for each category natively. How this algorithm works in detail is beyond the scope of this paper, but here is some intuition: A regression tree is fit to the data. A second regression tree is fit to the residuals from the first tree, a third tree is fit to the residuals of the sum of the first two trees, and so forth. How many such trees or iterations are needed? The number of trees is chosen such that classification all on a test data set is minimized and depends on the data and the values of the tuning parameters. That being said, thousands of trees are not unusual.

From a practitioner point of view it is important to know statistical learning algorithms are so flexible that they can produce (near) perfect prediction on any data set. This is called overfitting. To avoid overfitting, training data are usually split into two parts: the training data set is used to fit the model, and the test data set is used to evaluate whether the fit is good.

Most statistical learning algorithms have so-called tuning parameters that need to be set to reasonable values. For boosting, the following tuning parameters are often considered: degree of interactions (1, 2, 3, 4, 5, etc.), degree of bagging (typical values are 50%- 100%), and degree of shrinking (typical values are 0.1, 0.01, 0.001). In linear regression most scientists would not consider more than two-way interactions. In the context of boosting 5-way interactions would not be unusual. Shrinking refers to fitting the model with more iterations taking shrinking the step length at each iteration (i. e. taking smaller steps). Shrinking embodies the proverb "Slow and steady wins the race". A value of shrink=1 corresponds to full step length; 0.1 to one-tenth of the step length and so forth. Bagging refers to using only a subset of the data at each iteration. A value of 100% (bag=1.0) refers to using all the data at each iteration, whereas a bagging value of 50% (bag=0.5) means that a random half of the data are used at each iteration. This often leads to a more diverse model that may make it less dependent on individual variables. Experimenting with different values for tuning parameters can improve prediction on the test data.

Gradient boosting is implemented in Stata (Schonlau, 2005), the R language (Ridgeway, 2013), and elsewhere.

4 Semi-automatic categorization

Semi-automatic categorization involves five steps:

1. Manually categorize randomly selected training data (e. g. n=500). Rare categories with few observations in the training data (e. g. 10) can be combined into a single "rare" category. Develop a "correct" gold standard categorization for the training data. Typically, open-ended questions are categorized by two or more humans. To develop a gold standard from two or more categorizations, conflicts are reconciled either by consensus between the coders, by majority vote, or by an expert coder.

2. Turn the text answers (using both training and test data) into numerical variables.

3. Fit a statistical learning algorithm to the training data. In addition to categorization, the statistical learning algorithm needs to estimate the probability for each text falling into a category. In the examples below boosting is used; however, other algorithms could be used. Use the statistical learning algorithm to predict the category of uncategorized texts as well as the prediction probability.

4. Decide on a threshold probability (e. g., 0.8) for automatic categorization. The threshold should be chosen in view of the expected accuracy and the fraction of the test data that can be automatically categorized. "Accuracy" refers to the percentage of correct categorizations. Because it is unknown for uncategorized data, we rely on an estimate, the expected accuracy. The expected accuracy can be computed for the automatically categorized data only or of the combined ac-

Table 1
Hypothetical example of encoding three texts into variables (unigram and selected bigram indicators)

text	cat	eat	mouse	dog	bone	cat_eat	dog_eat	mouse_eat
The cat eats the mouse	1	1	1	0	0	1	0	0
The mouse eats the cat	1	1	1	0	0	0	0	1
The dog eats the bone	0	1	0	1	1	0	1	0

curacy of manual and automatic categorization. Appendix A gives formulas for estimates and variances as well as a lower bound for back-of-the-envelope calculations. The expected accuracy of the automatically categorized data is simply the average predicted probability (Equation 3 in Appendix A).

5. If the estimated categorization probability exceeds the threshold probability, then automatic categorization is accepted. If the estimated probability is lower than the threshold probability, manual categorization is required. A higher threshold corresponds to less automatic categorization with higher accuracy; a lower threshold corresponds to more automation with lower accuracy.

We evaluate the semi-automatic method against the alternative of coding all answers manually in terms of time savings and against the alternative of coding all answers automatically in terms of the percentage of correctly coded text answers. The percentage of correctly coded text answers is called accuracy.

5 Example 1: "Patient Joe" Question in Dutch

Consider the following open-ended question about a hypothetical scenario: "Joe's doctor told him that he would need to return in two weeks to find out whether or not his condition had improved. But when Joe asked the receptionist for an appointment, he was told that it would be over a month before the next available appointment. What should Joe do?" The original purpose of this question was to learn to what extent respondents would try to actively engage in the decision-making process and to what extent "patient activation" correlates with other variables such as literacy skills (Martin et al., 2011). Four code categories – proactive, somewhat proactive, passive and counterproductive – were articulated and a coding manual was developed. Coding reliability for two manual coders was $\kappa = 0.79$ (Martin et al., 2011). This question could not reasonably be asked as a single-response categorization question because the terms are not broadly understood and providing the terms would have biased the answers ("proactive" being more socially desirable than "counterproductive").

In 2012, the same question was asked in Dutch in the Internet panel LISS (http://www.lissdata.nl). Answers were also given in Dutch. Using the coding manual, two native Dutch speakers coded each of 1,758 responses into one of the 4 categories. Kappa was lower ($\kappa = 0.61$), presumably because the English version was coded by expert coders (investigators) and the Dutch version by non-expert coders (two students unrelated to the project). Differences were then resolved by an expert and yielded the "gold standard" categorization. Relative to the gold standard, the human coders were on average 87.7% accurate (the individual accuracies were 92.4% and 83.0%, respectively). The text answer contained a median of 17 words (Minimum: 1 word, 25th percentile: 10 words, 75th percentile: 97 words, maximum: 119 words).

Using the Stata program *ngram*, we created indicators of unigrams and bigrams that each appeared in at least 5 different answers (5 is the default in the software). We also included a variable that gave the length of the answer where length is defined as the number of words. We use Dutch language stemming and we removed Dutch stop words. This created more than 1,200 unigram and bigram variables plus the length-of-answer variable. We imported the data into Stata for boosting and other processing. We used 500 random observations as training data. We ran multinomial boosting with five-way interactions (interaction=5) and bagging (bag=0.5) and shrinking (shrink=0.1). Fitting a boosting model on the training data set with 500 observations took just over 10 minutes on a desktop computer (Intel Core i5-4590 chip and 8GB of RAM).

What fraction of the answers can be categorized automatically? It is possible to categorizing all answers automatically (threshold=0), but then the accuracy will be low. Expected values for accuracies and standard errors as a function of threshold can be computed (Appendix A). Results are summarized in Table 2. The choice of threshold value (first column) determines the fraction of answers that can be categorized automatically (second column) and the (estimated) expected accuracy among the automatically categorized answers (third column) along with the margin of error (half-width of a confidence interval) of that estimate (fourth column). For example, if a threshold of 0.7 is chosen, 47% of the data can be categorized automatically with an expected accuracy of 81% (±3.1%). The remainder, 53% of the data, would be categorized manually. Classifying all data automatically corresponds to a threshold of 0. If one were to categorize all data automatically, the achieved accuracy would be 69%.

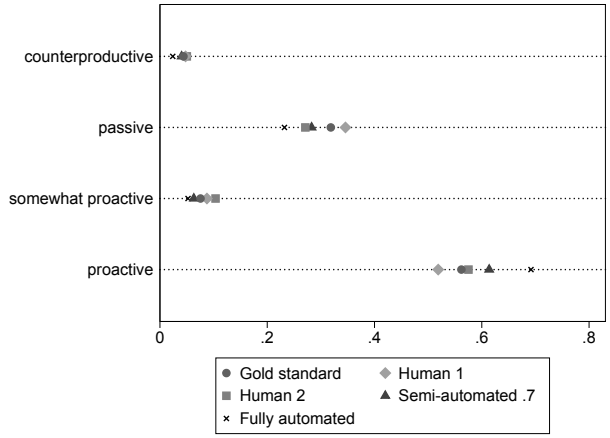

Figure 1. Distribution of the four categories based on the gold standard, human rater1, human rater2, semi-automated categorization with a threshold of 0.7, and fully automated categorization

To demonstrate the success of the method, not only the training data but also the test data were categorized manually. This makes it possible to compare the percentage of correctly classified texts ("Achieved Accuracy", fifth column) with the expected percentage based on the boosting model ("E(Accuracy)"). The expected accuracy (third column) matches the achieved accuracy (5th column) for automatic categorization very well, and is well within the margin of error.

Manual categorization is not 100% accurate either. Appendix A gives a formula for combining automatic and manual accuracies to get one overall measure of accuracy. The overall accuracy is always in between the expected accuracy and the (assumed) manual accuracy.

Figure 1 shows the distribution of the four categories based on each rater, the gold standard (with resolved differences between the two human raters), fully-automatic categorization and semi-automatic categorization with a threshold of 0.6 (the lowest threshold exceeding an expected accuracy of 0.8). The distribution of the semi-automatic categorization is roughly consistent with the distribution of two human categorizations. For the fully-automatic categorization, the percentage of texts categorized into the most frequent category "proactive" is somewhat larger than the two human categorizations.

6 Example 2: Risk of Disclosure

Couper, Singer, Conrad, and Groves (2008) investigated disclosure risks, privacy and confidentiality concerns as factors in survey participation. In a series of eight vignettes describing different surveys and with different risks of disclosure conditions, respondents were asked how likely they would be to take part or not take part in the survey described. After the first vignette, immediately following the "Willingness to participate" question, half the respondents were asked either a positively or negatively worded open-ended probe (Why would you participate? Why would you not participate?), depending on their response. The other half of the respondents received this question after the 8th vignette. Our analysis focuses here on the positively worded open-ended probe combining answers of questions asked after the 1st and 8th vignettes. 13.1% and 15.5% of respondents didn't answer the open-ended probe after the 1st and 8th vignettes, respectively. Open-ended probes were coded independently by two coders. Disagreements were reconciled by an expert coder. The interrater reliability was $\kappa = 0.79$ for the positively worded question.

The data contain 1,212 answers with 20 different categories. Some categories are rare with as few as 7 occurrences. Five hundred random observations were used as training data. All categories with less than 10 answers in the training data were categorized into a separate category "rare". This left 11 categories, including the "rare" category containing about 10% of the sample (Table 3). The text answers contained a median of 11 words (minimum: 1 word, 25th percentile: 5 words, 75th percentile: 18 words, maximum: 49 words). Using the Stata program *ngram* with English stemming and English stop words removed, we created a little over 1,000 unigram and bigram variables. The boosting parameters used were bagging=0.5, interaction=5, and shrink=0.1. This run took just under 11 minutes on a desktop computer (Intel Core i5-4590 chip and 8GB of RAM).

Expected accuracies for a range of threshold values are shown in Table 4. For example, if using a threshold of 0.6, then 58% of the data could be categorized automatically. For this threshold the expected overall accuracy is 0.81 (±0.037). If one categorizes all data automatically (threshold=0), the overall accuracy is 65%. This is much lower than the interrater reliability and not good enough if accuracy is the overriding concern.

As before, to demonstrate the success of the method we also categorized the test data manually. This allows a comparison of the expected accuracy with the percentage of correctly classified texts ("achieved accuracy"). The expected and achieved accuracies for the data that are to be categorized automatically are within 2 percentage points of each other and well within the margin of error.

The Disclosure data have substantially more categories (11 categories) than the Dutch Patient Joe data (4 categories). Nonetheless, the statistics in Table 4 are qualitatively comparable to the corresponding table for the Patient Joe data (Table 2). This suggests that an increase in the number of categories does not necessarily degrade overall accuracy. An increased number of categories also increases computer run time. The boosting algorithm fits one set of boosting trees

Table 2
Summary statistics for automatic categorization as a function of various thresholds for the Dutch "Patient Joe" data. Margin refers to the half width of a 95% confidence interval

Threshold	Fraction Auto Categorization	E(Auto Accuracy)	Margin Auto	Achieved Auto Accuracy
0.9	0.05	0.92	0.068	0.88
0.8	0.24	0.86	0.039	0.85
0.7	0.47	0.81	0.031	0.80
0.6	0.66	0.76	0.028	0.76
0.5	0.89	0.71	0.026	0.72
0	1.00	0.68	0.025	0.69

Table 3
Categories in the sample. Categories with less than 10 observations in the sample were combined (and are not listed)

		Training Data Size
1	Believes in research generally	16
2	Wants to be helpful / express opinion	144
3	General altruism	67
4	I'd learn something, interested in results, curiosity	16
5	The money (incentive)	67
6	Topic-related (interesting, issues important)	59
7	Survey doesn't take much time, short survey	23
8	Other survey characteristic	12
9	No objection	24
10	Uncodable response, ambiguous	25
11	Rare categories combined	47
		500

for each category; therefore running time increases linearly with the number of categories.

Figure 2 shows the distribution of all categories based on the gold standard, fully-automatic categorization and semi-automatic categorization with a threshold of 0.6 (the lowest threshold exceeding an expected accuracy of 0.8). The individual ratings of the human raters were not available. The distribution of the gold standard matches the distribution of the semi-automatic categorization very well. The fully automatic categorization again over-estimates the percentage of the most frequent category (2).

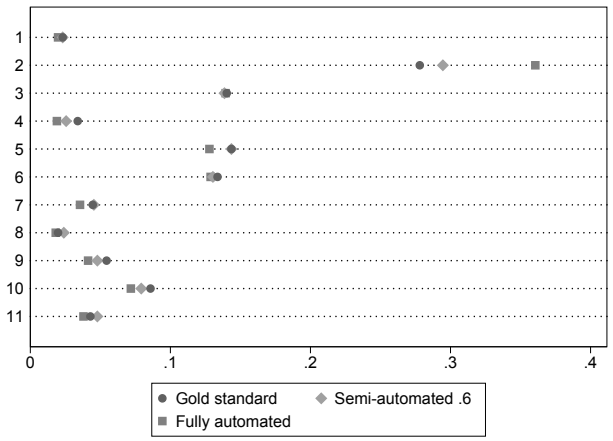

Figure 2. Distribution of the 11 categories based on the gold standard, semi-automated categorization with a threshold of 0.6, and fully-automated categorization. Category numbers match those in Table 3

7 Time Savings

Some rough calculations may clarify the potential for time savings over manual coding for this approach. Assuming a total of 1,500 open-ended answers, 500 answers to be used as training data, and half of the 1,000 answers in the test data can be categorized automatically, then 500 answers can be categorized automatically. For the Dutch data, manually categorizing 100 text answers took about 1.4 hours. In prac-

Table 4
Summary statistics for various thresholds for the Disclosure data. Margin refers to the half width of a 95% confidence interval

Threshold	Fraction Automatically Categorized	E(Accuracy)	Margin	Achieved Accuracy
0.9	0.15	0.94	0.045	0.95
0.8	0.31	0.90	0.040	0.90
0.7	0.46	0.85	0.038	0.87
0.6	0.58	0.81	0.037	0.82
0.5	0.70	0.76	0.035	0.76
0	1.00	0.65	0.031	0.65

tice, assuming two independent categorizations, automatically categorizing 500 observations would therefore save approximately $[(1,500-500)\cdot 0.5\cdot 2\cdot 1.4]/100 = 14$ hours. To the extent that human categorizations differ from one another, the two categorizations need to be reconciled either by a third human coder or by other means. This is not required for automated categorization and a modest amount of additional time could be saved. Assume a total of 10,000 open-ended answers, the same calculations would yield a time saving of $[(10,000-500)\cdot 0.5\cdot 2\cdot 1.4]/100 = 133$ hours or 16.6 eight-hour work days.

In the Disclosure data, categories with fewer than 10 answers in the training data were combined. To the extent that this combined category is predicted, answers in this category still need to be classified manually even when the probability is above the threshold. This reduces the time savings by about 10%. (In the Disclosure data 10% of the answers were classified in the combined "rare" category.)

Time savings also need to be weighed against the human time and expertise to create variables from the text answers and to run the statistical learning algorithm. Even taking the setup into account, these are substantial savings.

Computer running time for learning the boosting model is not typically a concern because the training data set is small for typical applications. Prediction does not increase running time much. If the size of the training data is the same, a data set with 10,000 observations will not take much longer to run than one with 1,000 observations. Running time increases in particular with small shrinkage values and with a larger number of categories. In our experience, running time of the Stata implementation of the boosting algorithm is generally under 2 hours with shrinkage as low as 0.01 with a couple dozen categories.

8 Discussion

Automatic categorization of open-ended questions is often not sufficiently accurate for analysts' needs. The proposed semi-automatic method for the categorization of open-ended questions requires fewer manual categorizations while still achieving a high level of accuracy. The expected accuracy can be controlled as a function of the threshold. To target an expected accuracy of 80%, thresholds of 0.7 and 0.6 were sufficient in the examples.

The distribution of categories obtained by the proposed procedure is comparable to that of human categorizations. In contrast, for full automation the most frequent category is sometimes over-predicted. Answer texts with high uncertainty are disproportionally assigned to the most frequent category which can distort the distribution.

The method lends itself in particular to Web surveys because the open-ended text is already in machine-readable format. This is also true for computer-assisted interviewing (CATI or CAPI) as the interviewer would have already transcribed the respondent's answer. The method is most useful for larger data sets or for questions where categorization is unusually time consuming.

The two examples were trained on 500 observations. This number will usually suffice for a training data set, though some more complex problems with many categories may require larger training data. For less complex problems, additional savings attained by reducing the training data set from 500 to a smaller number like 300 or 200 have to be weighed against the risk of a model not fitting as well. On balance, one might prefer a training data set a little larger to be on the safe side.

Choosing a meaningful threshold probability requires that predicted probabilities are approximately unbiased. Appendix B contains a simulation using different combinations of values for tuning parameters for all 3 examples. On average, predicted probabilities were approximately unbiased (Figure B1) when shrinkage is used (avoid shrink=1) and when higher order interactions are fit rather than just main effects (avoid interaction=1). Shrinkage is known to reduce the categorization error (Hastie, Tibshirani, & Friedman, 2009), but this is not the same as estimating approximately unbiased probabilities. Fitting 3-way and 5-way interactions also increases the fraction of data that can be automatically categorized.

In the computer science literature automatic coding of

open-ended questions using statistical learning is well known (Esuli, Fagni, & Sebastiani, 2010; Giorgetti & Sebastiani, 2003; Macer, Pearson, & Sebastiani, 2007). The computer science literature is mostly focused on multi-response categorization (Martinez-Alvarez, Bellogin, & Roelleke, 2013, 2012). In multi-response categorization a text can be classified into multiple categories (e. g. a movie can be both "independent" and a "drama"). In single-response categorization a text is classified in only one category (e. g., hair color cannot be both black and blond, the patient Joe response cannot be both "passive" and "proactive"). Multi-response questions have different challenges because there are multiple thresholds and prediction uncertainties.

In summary, it is possible to automatically categorize a portion of open-ended survey questions without compromising accuracy. This makes the proposed procedure preferable to fully-manual coding. Because of the additional time required for setting up the proposed method the number of text answers should be 1,500 or greater before meaningful time savings can be realized.

Acknowledgements

This research was supported by the Social Sciences and Humanities Research Council of Canada (SSHRC # 435-2013-0128). The disclosure project was supported with funds from the U.S. National Institute of Child Health and Human Development (NICHD grant #P01 HD045753-01 to Couper, Singer, Conrad and Groves). In this paper we make use of "Patient Joe" data of the LISS (Longitudinal Internet Studies for the Social sciences) panel administered by CentERdata (Tilburg University, The Netherlands). We thank the anonymous referees for their comments.

References

Couper, M. P., Singer, E., Conrad, F. G., & Groves, R. M. (2008). Risk of disclosure, perceptions of risk, and concerns about privacy and confidentiality as factors in survey participation. *Journal of Official Statistics*, *24(25)*, 255–275.

Esuli, A., Fagni, T., & Sebastiani, F. (2010). Machines that learn how to code open-ended survey data. *International Journal of Market Research*, *52(6)*, 775–800.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, *28(2)*, 337–407.

Geer, J. G. (1991). Do open-ended questions measure "salient" issues? *Public Opinion Quarterly*, *55(3)*, 360–370.

Giorgetti, D. & Sebastiani, F. (2003). Automating survey coding by multiclass text categorization techniques. *Journal of the American Society for Information Science and Technology*, *54(14)*, 1269–1277.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning (2nd edition)*. New York: springer.

Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. *LDV Forum*, *20(1)*, 19–62.

Joachims, T. (2002). *Learning to classify text using support vector machines: methods, theory and algorithms.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Macer, T., Pearson, M., & Sebastiani, F. (2007). Cracking the code: what customers say, in their own words. Proceedings of the 50th Annual Conference of the Market Research Society (MRS'07) in Brighton, UK.

Martin, L. T., Schonlau, M., Haas, A., Derose, K. P., Rosenfeld, L., Buka, S. L., & Rudd, R. (2011). Patient activation and advocacy: which literacy skills matter most? *Journal of Health Communication*, *16*(sub 3), 177–190.

Martinez-Alvarez, M., Bellogin, A., & Roelleke, T. (2013). *Document difficulty framework for semi-automatic text classification*. Proceedings of the 15th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2013), Prague, CZ.

Martinez-Alvarez, M., Yahyaei, S., & Roelleke, T. (2012). Semi-automatic document classification: exploiting document difficulty. In *Proceedings of the 34th european conference on information retrieval (ecir 2012), barcelona, es* (pp. 468–471).

Mayfield, E. & Penstein Rose, C. (2012). Lightside: text mining and machine learning user's manual. Carnegie Mellon University, Pittsburgh. Retrieved from http://www.cs.cmu.edu/~emayfiel/LightSIDE.pdf

Meyer, D., Hornik, K., & Feinerer, I. (2008). Text mining infrastructure in R. *Journal of Statistical Software*, *25(5)*, 1–54.

Ridgeway, G. (2013). Generalized boosted models: a guide to the gbm package. Retrieved from http://cran.at.r-project.org/web/packages/gbm/gbm.pdf

Schonlau, M. (2005). Boosted regression (boosting): an introductory tutorial and a Stata plugin. *The Stata Journal*, *5(3)*, 330–354.

Schonlau, M. & Guenther, N. (2016). Text mining using N-Grams. Retrieved from http://ssrn.com/abstract=2759033

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: practical machine learning tools and techniques*. Amsterdam: Elsevier.

Appendix A
Expected accuracy of categorized answers

We can compute the expected accuracy of manual and automatically categorized answers as a function of the threshold probability. Accuracy refers to the fraction of observations that are correctly coded. Neither manual coding nor automatic coding is 100% accurate. Denote the coding of text_i by

$$X_i = \begin{cases} 1 & \text{coding is correct} \\ 0 & \text{otherwise} \end{cases}$$

X_i follows an independent Bernoulli distribution with expected value $E(X_i) = p_i$ if text_i is coded automatically and $E(X_i) = p_{\text{man}}$ if text_i is coded manually, $i = 1, \ldots, n$ where n is the total number of texts. That means for automatically coded texts different texts are presumed to have different probabilities of being coded correctly; i. e. some answers are easier to code than others.

Then the expected percentage of correctly coded texts is

$$\begin{aligned} E(\bar{X}) &= \frac{1}{n}\left(\sum_{i \in S_{\text{man}}} p_{\text{man}} + \sum_{i \in S_{\text{auto}}} p_j\right) \\ &= \frac{n_{\text{man}}}{n} p_{\text{man}} + \frac{n_{\text{auto}}}{n}\left(\frac{1}{n_{\text{auto}}} \sum_{i \in S_{\text{auto}}} p_j\right) \\ &= \left(1 - \frac{n_{\text{auto}}}{n}\right) p_{\text{man}} + \frac{n_{\text{auto}}}{n} \bar{p}_{\text{auto}} \end{aligned} \quad (1)$$

where S_{man} and S_{auto} denote the set of texts which are manually and automatically coded, respectively, n_{man} and n_{auto} are the number of manually and automatically coded texts, respectively; and the sum of all coded texts is n, $n_{\text{man}}+n_{\text{auto}} = n$. The variance $Var(\bar{X})$ is as follows:

$$\begin{aligned} \text{Var}(\bar{X}) &= \frac{1}{n^2} \sum_{i=1}^{n} p_i(1 - p_i) \\ &= \frac{1}{n^2}\left(n_{\text{man}} p_{\text{man}}(1 - p_{\text{man}}) + \sum_{i=1}^{n_{\text{auto}}} p_i(1 - p_i)\right) \end{aligned} \quad (2)$$

and a 95% confidence interval can be computed as usual as $E(\bar{X}) \pm 1.96\sqrt{Var(\bar{X})}$. In the example data sets expected accuracy are compared with achieved accuracy for automatically coded texts. This is possible because in the examples presented all data were categorized manually (not just the training data). The expected accuracy for automatically coded texts is as follows

$$E(\bar{X}_{\text{auto}}) = \frac{1}{n_{\text{auto}}} \sum_{i \in S_{\text{auto}}} X_i = \bar{p}_{\text{auto}} \quad (3)$$

The corresponding variance is

$$Var(\bar{X}_{\text{auto}}) = \frac{1}{n_{\text{auto}}^2} \sum_{i \in S_{\text{auto}}} p_i(1 - p_i) \quad (4)$$

The achieved auto-accuracy (percentage of correctly categorized observations) is

$$\frac{1}{n_{\text{auto}}} \sum_{i=1}^{n_{\text{auto}}} X_i \quad (5)$$

Equation 2 uses estimates of individual probabilities that are obtained from the statistical learning algorithm. For back-of-the-envelope calculations that do not require the statistical learning algorithm, a lower bound on the expected percentage of correctly coded texts may be useful. Noticing $p_j \geq p_{\text{thres}}$ for all j in S_{auto}:

$$E(\bar{X}) \geq \left(1 - \frac{n_{\text{auto}}}{n}\right) \bar{p}_{\text{man}} + \frac{n_{\text{auto}}}{n} p_{\text{thres}} \quad (6)$$

In practice, the true categories are not known. Instead, a "gold standard" is constructed from multiple manual categorizations where any differences have been resolved (e. g. by an expert coder). If the "gold standard" were 100% accurate equation 6 becomes

$$E(\bar{X}) \geq \left(1 - \frac{n_{\text{auto}}}{n}\right) + \frac{n_{\text{auto}}}{n} p_{\text{thres}} = 1 - \frac{n_{\text{auto}}}{n}(1 - p_{\text{thres}}) \quad (7)$$

This means the combined accuracy depends only on the threshold probability and the fraction of texts that can be coded automatically. For example, if a threshold probability of 90% is assumed, and 50% of the data can be categorized automatically, then the lower bound for accuracy from 7 is $1 - 0.5(1 - 0.9) = 0.95$ or 95%. If manual categorization accuracy of 90% instead of 100% is assumed, the expected percentage of accurately coded texts based on equation 6 exceeds $0.5 \cdot 0.9 + 0.5 \cdot 0.9 = 0.9$ or 90%.

Appendix B
Sensitivity of boosting tuning parameters

Like all statistical learning techniques, boosting has some tuning parameters. Here we investigate whether the predicted probabilities are approximately unbiased and how the fraction of answers that can be predicted automatically varies for a range of values for tuning parameters. Approximately unbiased predicted probabilities is important for setting meaningful thresholds.

We conducted a factorial experiment with 3 values for shrinking (1 "no shrinking", 0.1, 0.01), 3 values for interactions (1 "main effects only", 3, 5), and 2 values of bagging (0.5, 1 "no bagging") and 15 replications for a total of $3 \cdot 3 \cdot 2 \cdot 15 = 270$ runs for each of the three data sets. Each replication corresponds to a different random set of the training data ($n = 500$).

Figure B1 displays box plots of the difference of predicted accuracy minus the achieved accuracy in the test data by different combinations of tuning parameters for all three examples. A difference of zero implies perfect prediction. Differences greater than zero imply conservative estimates

of the predicted accuracy; the achieved accuracy was larger. Boxplots centered on zero corresponding to unbiased estimates are preferred, those centered on positive values are conservative, and those centered on negative values are undesirable. Figure B1 shows that shrinking (0.1 or 0.01) is necessary. Among runs with shrinking, bagging (bag=0.5) tends to have little or no influence on whether the estimates are approximately unbiased.

The fraction of test data that can be automatically categorized using a threshold of 0.8 is shown in Figure B2 for the same combinations of shrinking, bagging, and interactions. Among the runs with shrinking, higher interactions (interaction=3, 5) and bagging (bag=0.5) tend to increase the fraction of test data that can be categorized automatically. Overall, taking into account both criteria, bagging (bag=0.5), shrinking (0.1 or 0.01) and greater interactions (3 or 5) may be preferable.

The fraction of the test data that can be categorized automatically ("automated fraction") varies substantially both within and between examples. The examples in sections 4 to 6 used the following tuning parameters: shrink=0.1; bag=0.5; interaction=5. For these parameter settings, the "automated fraction" in the simulation range anywhere from 20% to 70% for the Patient Joe data, and between 10% to 40% for the Disclosure data. This "automated fraction" is known after running the boosting algorithm and appropriate counter measures could be taken if the fraction is deemed too low. Such counter measures might include increasing the size of the training data.

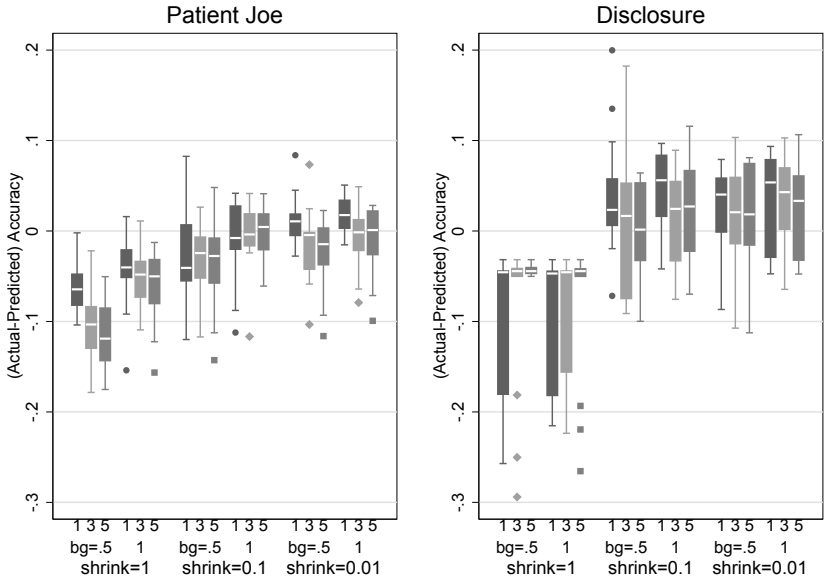

Figure B1. Box plots of the difference of predicted accuracy minus the achieved accuracy in the test data by different combinations of shrinking (3rd row), bagging (2nd row) and interaction (1st row)

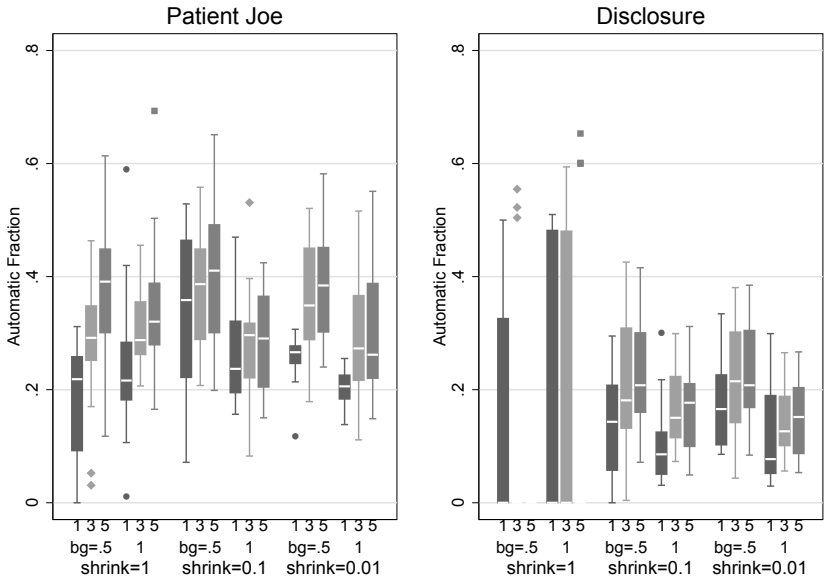

Figure B2. Box plots of the fraction of test data that can be automatically categorized using a threshold of 0.8 by different combinations of shrinking (3rd row), bagging (2nd row) and interaction (1st row)