methods, data, analyses | Vol. 15(1), 2021, pp. 103-120 DOI: 10.12758/mda.2020.10

Coding Text Answers to Open-ended Questions: Human Coders and Statistical Learning Algorithms Make Similar Mistakes

Zhoushanyue He & Matthias Schonlau
University of Waterloo

Abstract

Text answers to open-ended questions are often manually coded into one of several predefined categories or classes. More recently, researchers have begun to employ statistical models to automatically classify such text responses. It is unclear whether such automated coders and human coders find the same type of observations difficult to code or whether humans and models might be able to compensate for each other's weaknesses. We analyze correlations between estimated error probabilities of human and automated coders and find: 1) Statistical models have higher error rates than human coders 2) Automated coders (models) and human coders tend to make similar coding mistakes. Specifically, the correlation between the estimated coding error of a statistical model and that of a human is comparable to that of two humans. 3) Two very different statistical models give highly correlated estimated coding errors. Therefore, a) the choice of statistical model does not matter, and b) having a second automated coder would be redundant.

Keywords: open-ended question, manual coding, automatic coding, text classification, text answer

Open-ended questions yield text data. This makes them hard to analyze with quantitative methods. Often, open-ended responses are coded into pre-specified codes (or categories or classes). Researchers classify text answers either manually or automatically.

Manual coding refers to human coders classifying text answers, usually based on a coding manual. To the extent that some or all of the data are coded by two coders, any differences need to be resolved (e.g., with an expert coder, or by employing a third coder). We call the resulting resolved code the gold standard code.

Automatic coding refers to using a statistical learning model (or "automated coder") to predict the code of text answers. Automatic coding still requires a manually coded smaller training data set: First, a randomly selected subset of the data is selected as training data and coded manually. The size of the training data can vary but would typically consist of a few hundred answer texts. Second, the answer texts of all answers are converted into numerical n-gram variables (see section "Background"). Third, a statistical learning model is trained on the training data set. Typically, the gold standard codes are used for training. (For other approaches see He & Schonlau, to appear). Fourth, the statistical learning algorithm predicts the most likely code.

Both human and automatic coding make mistakes but for different reasons. Manual coding error stems from human error, ambiguous text answers, and an unclear coding manual. Automatic coding makes mistakes because of statistical generalization error and because of any remaining coding mistakes in the gold standard codes. While the reasons for mistakes are different, it is unclear whether the automatic coding makes similar mistakes as human coders. For example, we do not know whether a text answer that is difficult for human coders is also difficult for automated coders, or whether automated coders work well on a text answer that human coders find easy to code.

There is no reason to believe that humans and automated coders necessarily make similar mistakes: a statistical learning algorithm cannot reason like a human. A learning algorithm based on so called n-gram variables evaluates the presence or absence of words, or the number of times a word appears, whereas humans try to understand entire sentences.

Acknowledgements

This research was supported by the Social Sciences and Humanities Research Council of Canada (SSHRC # 435-2013-0128). We thank Dr. Katharina Meitinger at the University of Utrecht for allowing us to use the Happiness and Democracy data.

Direct correspondence to

Matthias Schonlau, University of Waterloo, 200 University Ave W,
Waterloo ON N2L 3G1, Canada
E-mail: schonlau@uwaterloo.ca

This paper explores to what extent human coders and automated coders make similar coding mistakes. The outline of this paper is as follows: The next section introduces background on manual coding and automatic coding for open-ended questions. The third section describes the datasets and automatic coding methods we use in this paper. The fourth section investigates similarities and differences between human and automatic coding. The last section discusses conclusions and limitations.

Background

Open-ended questions are particularly useful if researchers do not want to constrain respondents' answers to pre-specified selections. Open-ended questions allow respondents to provide diverse answers based on their experience, and some answers are probably never thought of by researchers. For example, Bengston et al. (2011) found an open-ended question revealed diverse and multidimensional motivations expressed by respondents, while closed-ended question failed to capture many dimensions.

Text data from open-ended questions are usually more difficult for quantitative analysis than numeric data because they are unstructured. A common way of analyzing text data is to classify them into classes/categories, either manually or automatically. Usually, text answers are coded manually using human coders (Roberts et al., 2014). A disadvantage of manual coding is that it tends to be expensive (Geer, 1991; Grimmer & Stewart, 2013). Moreover, the manual coding process is subjective (Patel et al., 2012), whereas automatic coding is not. For large data sets, automatic coding is also more cost-efficient (Chai, 2019).

Statistical learning enables automatic text classification. Popular statistical learning methods applied in analyzing open-ended questions include Naïve Bayes (Severin et al., 2017), support vector machines (Bullington et al., 2007) and tree-based methods (random forests, boosting) (Kern et al., 2019). Some researchers have combined statistical learning algorithms with manual coding to achieve better classification. For example, Schonlau & Couper (2016) proposed a semi-automatic algorithm based on multinomial gradient boosting to code text answers automatically if automatic coding was likely to be correct or code manually otherwise.

Both human coders and statistical models make mistakes, yet the sources of mistakes may be different. Humans make mistakes because of the ambiguity of texts, fatigue, unclear codebooks or a misunderstanding of the meaning of responses (Funkhouser & Parker, 1968; He & Schonlau, to appear). Conrad et al. (2016) have examined the misclassification of open occupation descriptions and found that longer descriptions are less reliably coded than shorter descriptions for easy occupation terms, but slightly more reliably coded for difficult occupation

terms. Researchers have emphasized the need to assess and improve coder reliability (Crittenden & Hill, 1971; Kassarjian, 1977; Montgomery & Crittenden, 1977; Hughes & Garrett, 1990). Lombard et al. (2002) provided a standard guideline for assessing and reporting inter-coder reliability. Coding error of automated coders include human error in the training data (Belloni et al., 2016) and generalization (out of sample) error of the fitted model (Giorgetti et al., 2003). If the data are double coded the resulting gold standard codes should have little remaining human error. The primary source of coding error for automated coding is generalization error.

Statistical learning algorithms expect numerical data. Answer texts have to be converted to numerical variables. n-gram variables with n=1 contain counts or indicators of how often a given word occurs in a text. n-gram variables with n=2 contain counts or indicators of how often a given word sequence of two words occurs in a text. As each unique word is turned into a variable, the number of variables is potentially very large. Additionally, a "number of words" variable that captures the length of the text answer is useful in almost all applications. Techniques exist to limit the number of variables (stemming, thresholds, stopwords) somewhat (Büttcher et al., 2016; Schonlau et al., 2017). Nonetheless, a regression with large number of variables requires flexible statistical learning methods, more flexible than logistic or multinomial regression.

Despite the widespread application of statistical learning, there are relatively few studies about classifying text answers from open-ended questions using statistical learning models. Conway (2006) pointed out that using automatic coding allowed researchers to avoid problems with inter-coder reliability, a major issue of human coding when multiple coders are involved. To the best of our knowledge, whether humans and models make similar coding errors has not yet been addressed in the literature.

Data and Statistical Learning Models

We use three double-coded datasets that we label the Patient Joe, Happiness and Democracy datasets. The size of these datasets as well as their percentage of inter-coder disagreement is listed in Table 1.

The Patient Joe dataset (Schonlau, 2020) contains answers to an open-ended question in a study fielded in Dutch in the LISS panel (http://www.lissdata.nl) in 2012. The question was to investigate patients' decision making by asking "Joe's doctor told him that he would need to return in two weeks to find out whether 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?" (Martin et al., 2011). These text answers were double coded

Table 1 The data size, the percentage of disagreement and kappa of the Patient Joe, Happiness and Democracy data.

	Size of the (whole) dataset	Size of training dataset	Size of test dataset	Percentage of disagreement	Kappa
Patient Joe	1756	1000	756	23.18%	0.61
Happiness	1438	800	638	5.77%	0.93
Democracy	1096	600	496	14.42%	0.82

by two coders into four classes: proactive, somewhat proactive, passive and counterproductive. The disagreement between the two coders was resolved by an expert.

Both the Happiness and Democracy datasets were collected in a web survey conducted in November 2017. The participants were from an online-access panel in Germany provided by respondi (http://www.respondi.com/EN/). The Happiness dataset contains responses to an open-ended question "What aspects of your life have you considered when assessing your happiness?" The data were classified into 10 classes such as social network & surrounding, health and job. The Democracy dataset contains responses to a probe question "What aspects did you think of when answering the question how satisfied you were with the way democracy works in Germany?" The data were classified into 7 classes such as "actors & groups", "public policy areas" and "evaluation of behavior of politicians & parties". Both datasets were double coded with inter-coder disagreement being resolved through a group discussion.

We use two widely used statistical learning models, support vector machines (SVM) and random forests (RF) as representatives of statistical learning models (James et al., 2013). SVM and RF are supervised learning methods like logistic or linear regression. However, they are far more flexible and usually predict better. SVMs are formulated as an optimization problem: For a binary outcome, SVMs find the separating hyperplane between the two classes that maximize the distance of the closest point to the hyperplane. Because the two outcome classes are almost never perfectly separable, an error budget allows for a certain amount of misclassification. Random forests take a very different approach: Broadly speaking, RF aggregate predictions from individual regression trees trained on bootstrap samples.

We randomly split each of the three datasets into a training dataset and a test dataset. The SVM and random forests are trained on the "gold standard coding" (the coding after disagreement-resolution) of the training data. We use the trained

models to predict the codes of the test data. These predicted codes are then referred as the codes of automated coders in later experiments.

Results

Do Automated Coders Achieve Similar Coding Accuracy as Human Coders?

Figure 1 shows the coding accuracy of two automated coders and two human coders in the three datasets. The coding accuracy is the proportion of codes that match the gold standard code. Earlier we said that automatic coding makes mistakes because of statistical generalization error and because of any remaining coding mistakes in the gold standard codes. When comparing to the gold standard code, the coding error of automated coding is only due to statistical generalization error, not due to human error. The coding accuracy is evaluated on the test data, as is appropriate for statistical learning models.

We see from Figure 1 that the coding accuracy of SVM and RF is lower than that of human coders. The differences are statistically significant in a two-proportion z-test: all p-values are smaller than 0.01. Therefore, when we investigate whether models and humans make the same mistake, we have to remove the effect of different error rates.

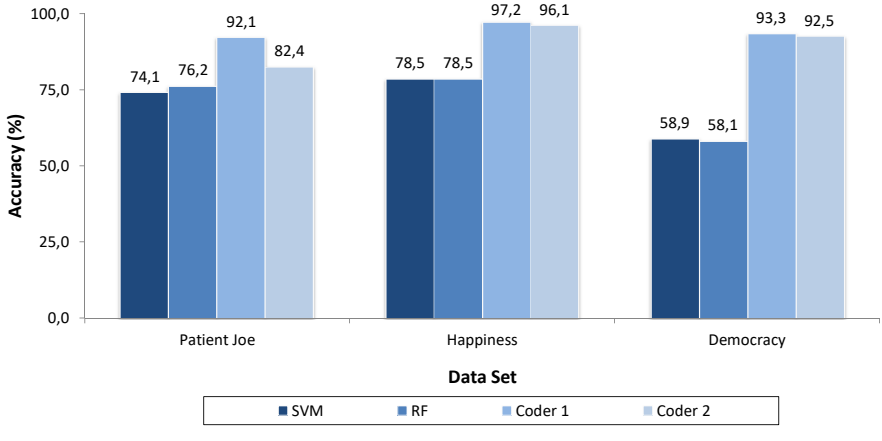

Figure 1 Coding accuracy of automated coders and human coders on the test data for the Patient Joe, Happiness and Democracy datasets.

Do Automated Coders and Human Coders have Similar Error Probabilities?

If both automated coders and human coders have a high probability to code an observation incorrectly, we infer that they make similar mistakes. Automated coders naturally produce the model-based probability of making a coding error. For example, suppose a model outputs the probability of a response belonging to one of four categories as follows: 0.6 "proactive", 0.2 "somewhat proactive", 0.1 "passive", and 0.1 "counterproductive". In that case the predicted category is "proactive". The model-based probability of an error depends on the true class of the response. If the true class is "proactive", the model-based error probability is 1-0.6=0.4 or 40%.

By contrast, human coders simply code an observation. The code is either correct (coded as 1) or incorrect (coded as 0). A model-based error probability is not available for human coders. However, we can estimate such a probability by aggregating the data into subsets. The estimated probability is then the proportion of correctly coded codes for each subset. Rather than forming the subsets at random, we order the observations by their average estimated model-based coding error probability. For example, if 10 subsets are desired, each decile of the observations ordered by their coding error probability forms one subset. Appendix A briefly illustrates this idea. In this paper, we divide the test set into 36 subsets for the Patient Joe dataset, 29 subsets for the Happiness dataset, and 31 subsets for the Democracy dataset.

Next, we compute two-way correlations among the estimated probabilities for the four coders (two automated coders and two human coders) for each dataset. Since the estimated coding error probabilities for humans only exist at the aggregated level, we also estimate the coding error probabilities for automated coders in each subset to make sure the probabilities of different coders are comparable. Table 2 shows a correlation matrix of estimated coding error probabilities.

We find that all the correlations are positive, and the correlation between an automated coder and a human coder is similar in magnitude to the correlation between two human coders. This suggests that both the human coders and the automated coders find the same observations easy or hard to code. Also, the extent of agreement between a human coder and an automated coder as compared to two human coders is very similar. However, the correlations only imply a tendency to find the same observations difficult; they do not imply the same level of accuracy. The previous section already found that human coders are more accurate as compared to automated coders.

We also find that the correlation between the two automated coders is very high. In fact, for the Democracy and Happiness data, the correlation rounds to 1.00. Given that the two automated coders also have almost the same accuracy (Figure 1), it does not matter which statistical learning model we choose: they are func-

Table 2 Correlation matrix of estimated error probabilities for each dataset.

	SVM	RF	Coder 1	Coder 2
Patient Joe				
SVM	1.00	0.95	0.44	0.88
RF		1.00	0.44	0.89
Coder 1			1.00	0.29
Coder 2				1.00
Happiness				
SVM	1.00	1.00	0.70	0.69
RF		1.00	0.71	0.69
Coder 1			1.00	0.65
Coder 2				1.00
Democracy				
SVM	1.00	1.00	0.53	0.31
RF		1.00	0.51	0.31
Coder 1			1.00	0.40
Coder 2				1.00

tionally equivalent. This is different for the two human coders which have a more moderate positive correlation.

The analysis of the correlation matrices reveals pairwise similarities for the four coders, yet the overall similarities or differences of the four coders is unclear. To answer this question, we use principal component analysis (PCA) to analyze the estimated error probabilities. The error probabilities of each of the four coders are standardized as part of PCA; standardization to the same mean removes the differential error rates among coders. The correlations between the coding error probabilities for each method and the principal components are listed in Table 3.

The three analyses of the three datasets tell similar stories. The first principal component explains most of the variation (65%-80%) in the estimated error probabilities among the four coders. The first principal component can be interpreted as an average of the four coders and represents what the coders have in common. The principal component corresponding to the difference between automated coders and human coders (the third component for the Patient Joe and the second component for the Happiness and Democracy data) explains 22% or less of the total variation. The remaining (second or third) principal component represents a specific

Table 3 Correlation between principal components and the original estimated error probabilities. The percentage of variation explained for each principal component is also given.

	Dim.1	Dim.2	Dim.3	Dim.4
Patient Joe				
SVM	0.97	0.10	0.18	0.15
RF	0.97	0.11	0.11	-0.17
Coder 1	0.55	-0.83	-0.05	0.00
Coder 2	0.92	0.28	-0.27	0.03
Variation explained	76.0%	19.7%	2.9%	1.3%
Happiness				
SVM	0.95	0.30	0.05	0.04
RF	0.95	0.29	0.04	-0.04
Coder 1	0.85	-0.27	-0.46	0.00
Coder 2	0.84	-0.41	0.37	-0.00
Variation explained	80.7%	10.4%	8.8%	0.1%
Democracy				
SVM	0.94	0.32	0.14	0.03
RF	0.93	0.33	0.16	-0.03
Coder 1	0.75	-0.25	-0.62	-0.00
Coder 2	0.55	-0.77	0.33	0.00
Variation explained	65.0%	21.7%	13.3%	0.1%

contrasts of one human coder vs. the other human coder and the two automated coders. The fourth principal component explains almost no variation because the two automated coders give nearly identical estimates, removing one dimension. In summary, the coders' estimated error probabilities exhibit far more communalities than differences.

Examples on Which Automated Coders and Human Coders Agree or Disagree

In an effort to gain further insight into differences and similarities between human coding and automatic coding, we now look at some specific coding examples for

one of the datasets, the Patient Joe data. The responses we discuss below are summarized in Table 4 with their English translation.

Some responses are inherently easy to code for both human and automated coders. For example, a response "I would accept." ("ik zou accepteren") is short and clear. Other responses appear more complicated, yet both human and automated coders code correctly. For example, the response "Feedback to the relevant physician. If Joe would get again nothing in response to the request (so only to have the possibility of an appointment in a month), request a second opinion from another doctor / hospital. This example happened to me!" is relatively long and consists of three sentences, but both human coders and automated coders correctly coded this response to be "proactive". Here "proactive" means that the patient insists on checking with the doctor rather than accepting the appointment or to go to another doctor/hospital. The categorization is not trivial for an automated coder, because the phrase "other doctor" is part of the respondent's answers. This suggests that automated coders can work well on both simple and complicated text answers.

The text is coded into n-gram variables, specifically indicator variables of the presence or absence of single words (unigrams) or bigrams. As a consequence, if individual n-gram variables are highly indicative of a code (or class) then the model will be able to code the text more easily. For example, in the Patient Joe data, if a response contains the phrase "2 weeks", the SVM or random forests model is likely to code it as "proactive" because most responses containing "2 weeks" say Joe should insist to see the doctor in two weeks. Highly discriminative n-gram variables often help automated coders, but not always. For example, a response "tell the assistant that he has to come again with 2 weeks and that there is probably still a place available" contains the words "2 weeks". However, such a response is not categorized as proactive in this coding scheme because merely telling the receptionist (rather than insisting/ refusing to accept) leaves a reasonable chance of failure. While both human coders realized this response is not proactive, the two automated coders still classified it as proactive because they relied on the words "2 weeks" too heavily. We understand that statistical models make complex tradeoffs between the variables and do not merely sum the evidence from each n-gram. Nonetheless, they are greatly helped by a few strong indicators.

Human coders and automated coders have different ways of dealing with responses that contain only new words not observed in the training data. Automated coders, once trained, assign these responses to a code based on the length of the response and the absence of all known words. In our experiments, the default code of SVM and random forests in the Patient Joe is "passive" for a response with 7 words, in the Happiness is "social network & surrounding" for a response with 2 words, and in the Democracy is "situation" for a response with 2 words. Human coders do not classify new responses only based on past coding experience; instead, they code using their knowledge. They can classify responses that are com-

Table 4 Example responses for various human vs. automatic coding results in the Patient Joe data. We show both the original response in Dutch and our English translation.

Coding result	Original response	Translated response
Human coders correct; automated coders correct. (short and easy)	ik zou accepteren.	I would accept.
Human coders correct; automated coders correct. (long and complicated)	Terugkoppelen naar de betreffende arts. Als Jan opnieuw nul op het request zou krijgen (dus alleen bij de mogelijkheid van een afspraak over een maand terecht zou kunnen), een second opinion aanvragen bij een andere arts / ziekenhuis Dit voorbeeld is mijzelf overkomen!	Feedback to the relevant physician. If Joe would get again nothing in response to the request (so only to have the possibility of an appointment in a month), request a second opinion from another doctor / hospital. This example happened to me!
Human coders correct; automated coders correct. (contains phrase "2 weeks")	Er op staan dat er toch over 2 weken een afspraak komt omdat ook de arts dit zo wil	Insist that there will be an appointment in 2 weeks because the doctor also wants this
Human coders incorrect; automated coders correct. (contains phrase "2 weeks")	zeggen tegen de assistente dat ie met 2 weken weer moet komen en dat er vast nog een plekje vrij is	tell the assistant that he has to come again with 2 weeks and that there is probably still a place available
Human coders correct; automated coders incorrect. (contains no known information)	thuis blyven	stay home

pletely new to any of the classes. For example, "stay home" ("thuis blyven") does not appear in the training data. SVM and random forests incorrectly classified it to the default code 2 (passive). By contrast, the human coders correctly classified the response to the code "counterproductive".

Discussion

We have investigated the relationship between automatic coding and manual coding by examining the similarities between their estimated coding errors. Crucially, we were able to estimate human coding error probabilities by aggregating the coded text answers to subsets. We found that when coding all observations automatically, automatic coding has a higher error rate than manual coding. However, coding errors correlate: automated coders and human coders tend to find the same responses difficult to code.

Although we find that human coders and automated coders make similar coding mistakes, the logic behind their mistakes is different. Automated coders code well on responses containing crucial words (unigrams or bi-grams): these words are usually indicators of some classes. These words may also help human coders, yet they are not as important as for automated coders (or humans can better understand responses containing no crucial words). Automated coders code responses without crucial words or without any known information by classifying them into the same default class (for a given answer length). Human coders do not have a default class: they code new responses based on understanding the meaning of texts.

The error rate is overall higher for automated coders based on n-gram variables than for human coders. Semi-automatic coding (Schonlau & Couper, 2016) – coding easy-to-code observations automatically and the remainder manually – is thus useful.

As is customary, the statistical learning models are trained on a random training subset of the data and predicted on the remaining test data. To confirm that the findings do not depend on the particular random train/test split, we also used leave-one-out cross validation and obtained qualitatively the same results.

Limitations of this study include: 1) We used SVM and random forests as representatives of automated coders. There are other statistical learning models. We believe that using a different model would not have large impacts on the results, which is partially demonstrated by the high similarity between SVM and random forests. 2) We estimated the error probability of human coders by dividing the data into multiple subsets and estimating the error in each subset. The estimation depends on the how we divide the data into subsets. We ordered observations based on the average error probabilities of SVM and random forests. This is not the only way of creating subsets but is preferable over random subsets in which the average probabilities would cluster more around the population mean.

In summary, automated coders and human coders tend to find the same text answers difficult to code. There is no point in having two different automated coders (RF and SVM): Automated coders almost always predict the same code.

Reference

Belloni, M., Brugiavini, A., Meschi, E., & Tijdens, K. (2016). Measuring and detecting errors in occupational coding: an analysis of share data. *Journal of Official Statistics*, *32*(4), 917–945. https://doi.org/10.1515/JOS-2016-0049

Bengston, D. N., Asah, S. T., & Butler, B. J. (2011). The diverse values and motivations of family forest owners in the United States: an analysis of an open-ended question in the national woodland owner survey. *Small-Scale Forestry*, *10*(3), 339–355. https://doi.org/10.1007/s11842-010-9152-9

Bullington, J., Endres, I., & Rahman, M. A. (2007). Open-ended question classification using support vector machines. In *Modern AI and Cognitive Science Conference (MAICS) 2007.* www.jrbcs.com/files/OE_Question_Classification_Using_SVM.pdf

Büttcher, S., Clarke, C. LA, & Cormack, G. V. (2016). *Information retrieval: Implementing and evaluating search engines*. MIT Press.

Chai, C. P. (2019). Text mining in survey data. *Survey Practice*, *12*(1), 1–14. https://doi.org/10.29115/sp-2018-0035

Conrad, F. G., Couper, M. P., & Sakshaug, J. W. (2016). Classifying open-ended reports: factors affecting the reliability of occupation codes. *Journal of Official Statistics*, *32*(1), 75–92. https://doi.org/10.1515/JOS-2016-0003

Conway, M. (2006). The subjective precision of computers: a methodological comparison with human coding in content analysis. *Journalism and Mass Communication Quarterly*, *83*(1), 186–200. https://doi.org/10.1177/107769900608300112

Crittenden, K. S., & Hill, R. J. (1971). Coding reliability and validity of interview data. *American Sociological Review*, *36*(6), 1073–1080. https://doi.org/10.2307/2093766

Funkhouser, G. R., & Parker, E. B. (1968). Analyzing coding reliability: the random-systematic-error coefficient. *Public Opinion Quarterly*, *32*(1), 122–128. https://doi.org/10.1086/267585

Geer, J. G. (1991). Do open-ended questions measure "salient" issues? *Public Opinion Quarterly*, *55*(3), 360–370. https://doi.org/10.1086/269268

Giorgetti, D., Prodanof, I., & Sebastiani, F. (2003). Automatic coding of open-ended questions using text categorization techniques. *Proceedings of the 4th International Conference of the Association for Survey Computing (ASCIC 2003)*, 173–184.

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, *21*(3), 267–297. https://doi.org/10.1093/pan/mps028

He, Z., & Schonlau, M. (n.d.). Automatic coding of text answers to open-ended questions: should you double code the training data? *Social Science Computer Review*. https://doi.org/10.1177/0894439319846622

Hughes, M. A., & Garrett, D. E. (1990). Intercoder reliability estimation approaches in marketing: a generalizability theory framework for quantitative data. *Journal of Marketing Research*, *27*(2), 185–195. https://doi.org/10.2307/3172845

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

Kassarjian, H. H. (1977). Content analysis in consumer research. *Journal of Consumer Research*, *4*(1), 8–18. https://doi.org/10.1086/208674

Kern, C., Klausch, T., & Kreuter, F. (2019). Tree-based machine learning methods for survey research. *Survey Research Methods*, *13*(1), 73–93. https://doi.org/10.18148/srm/2019.v13i1.7395

Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2002). Content analysis in mass communication: assessment and reporting of intercoder reliability. *Human Communication Research*, *28*(4), 587–604. https://doi.org/10.1093/hcr/28.4.587

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*(SUPPL. 3), 177–190. https://doi.org/10.1080/10810730.2011.604705

Montgomery, A. C., & Crittenden, K. S. (1977). Improving coding reliability for open-ended questions. *Public Opinion Quarterly*, *41*(2), 235–243. https://doi.org/10.1086/268378

Patel, M. D., Rose, K. M., Owens, C. R., Bang, H., & Kaufman, J. S. (2012). Performance of automated and manual coding systems for occupational data: a case study of historical records. *American Journal of Industrial Medicine*, *55*(3), 228–231. https://doi.org/10.1002/ajim.22005

Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., Albertson, B., & Rand, D. G. (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science*, *58*(4), 1064–1082. https://doi.org/10.1111/ajps.12103

Schonlau, M. (2020). Size text box, Patient Joe data. *CentERdata*. https://www.dataarchive.lissdata.nl/study_units/view/971

Schonlau, M., & Couper, M. P. (2016). Semi-automated categorization of open-ended questions. *Survey Research Methods*, *10*(2), 143–152. https://doi.org/10.18148/srm/2016.v10i2.6213

Schonlau, M., Guenther, N., & Sucholutsky, I. (2017). Text mining with n-gram variables. *The Stata Journal*, *17*(4), 866–881. https://doi.org/10.1177/1536867X1801700406

Severin, K., Gokhale, S. S., & Konduri, K. C. (2017). Automated quantitative analysis of open-ended survey responses for transportation planning. *2017 IEEE SmartWorld Ubiquitous Intelligence and Computing, Advanced and Trusted Computed, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017 -*, 1–7. https://doi.org/10.1109/UIC-ATC.2017.8397567

Appendix A

An Example of How to Estimate Error Probabilities of Human Coders

Suppose a model classifies an observation correctly (based on the gold standard code) with a probability of, for example, 0.7. Then the model-based error probability is 0.3. Humans just choose a code; no error probability is available. In this appendix we illustrate how the error probabilities of human coders are estimated using a toy data set. Table A1 shows the error probabilities of automated coders and whether the codes of the human coder are correct based on the gold standard code (the columns of Coder 1 and Coder 2). For this example, only the models' error probability matters; what code SVM and RF chose is not relevant.

Table A1 Model-based error probabilities and whether or not human coders coded correctly based on the gold-standard in the toy example.

Observation Index	SVM error probability	RF error probability	Coder 1	Coder 2
1	0.1	0.2	correct	correct
2	0.1	0.1	correct	correct
3	0.3	0.2	incorrect	correct
4	0.5	0.3	correct	incorrect
5	0.2	0.4	incorrect	incorrect
6	0.1	0.0	correct	correct
7	0.6	0.4	incorrect	incorrect
8	0.2	0.4	correct	incorrect
9	0.3	0.3	correct	correct
10	0.5	0.4	incorrect	incorrect
11	0.2	0.1	correct	correct
12	0.2	0.2	incorrect	correct
13	0.3	0.2	correct	correct
14	0.2	0.3	correct	correct
15	0.4	0.5	incorrect	correct

First, we compute the average error probability of the two automated coders (SVM and RF). We then sort the observations according to the average error probability. Next, we divide the ordered observations into equal-sized subsets. In this example, we choose 3 subsets: A, B and C. Table A2 shows the grouping of observations.

Table A2 Observations ordered by the average error probability of the automated coders in the toy example.

Observation Index	Coder 1	Coder 2	SVM error prob.	RF error prob.	Average error probability of automated coders	Subset
7	incorrect	incorrect	0.6	0.4	0.5	A
10	incorrect	incorrect	0.5	0.4	0.45	A
15	incorrect	correct	0.4	0.5	0.45	A
4	correct	incorrect	0.5	0.3	0.4	A
5	incorrect	incorrect	0.2	0.4	0.3	A
8	correct	incorrect	0.2	0.4	0.3	B
9	correct	correct	0.3	0.3	0.3	B
3	incorrect	correct	0.3	0.2	0.25	B
13	correct	correct	0.3	0.2	0.25	B
14	correct	correct	0.2	0.3	0.25	B
12	incorrect	correct	0.2	0.2	0.2	C
1	correct	correct	0.1	0.2	0.15	C
11	correct	correct	0.2	0.1	0.15	C
2	correct	correct	0.1	0.1	0.1	C
6	correct	correct	0.1	0.0	0.05	C

Next, we compute the human error probabilities within each subset. Among the 5 observations in subset A, coder 1 matches the gold standard codes on one observation only. Therefore, we estimate the error probability of coder 1 on subset A as 1-1/5=0.8 or 80%. Similarly, in subset B, coder 1 matches the gold standard codes on four observations, and the estimated error probability of coder 1 on subset B is 1-4/5=0.2 or 20%. We compute the remaining human error probabilities analogously. For automated coders, we average the error probabilities within each subset. The averaged error probability of automated coders and the estimated error probability of human coders per subset are shown in Table A3.

Table A3 Average error probabilities of human and automated coders for each subset.

Subset	Error probability of Coder 1	Error probability of Coder 2	Average error probability of SVM	Average error probability of RF
A	0.8	0.8	0.44	0.4
B	0.2	0.2	0.26	0.28
C	0.2	0	0.14	0.12