Automatic Coding of Text Answers to Open-ended Questions: Should You Double Code the Training Data?

Zhoushanyue He
University of Waterloo

Matthias Schonlau
University of Waterloo

Abstract

Open-ended questions in surveys are often manually coded into one of several classes (or categories). When the data are too large to manually code all texts, a statistical (or machine) learning model must be trained on a manually coded subset of texts. Uncoded texts are then coded automatically using the trained model. The quality of automatic coding depends on the trained statistical model, and the model relies on manually coded data on which it is trained. While survey scientists are acutely aware that the manual coding is not always accurate, it is not clear how double-coding affects the classification errors of the statistical learning model. We investigate several budget allocation strategies when there is a limited budget for manual classification: single-coding versus various options for double-coding where

the number of training texts is reduced to maintain the fixed budget. Under fixed budget, double-coding improved prediction of the learning algorithm when the coding error is greater than about 20%-35%, depending on the data. Among double-coding strategies, paying for an expert to resolve differences performed best. When no expert is available, removing differences from the training data outperformed other double-coding strategies. When there is no budget constraint and the texts have already been double coded, all double-coding strategies generally outperformed single-coding. As under fixed budget, having an expert to solve disagreement in training texts improves accuracy most, followed by removing differences.

Keywords: Double-coding, Statistical learning, Machine learning, Open-ended questions, Manual coding, Text classification, Human coder

1 Introduction

Traditionally, text data collected from open-ended questions in surveys have been manually classified at great expense. Recently, automatic classification of text data from open-ended questions or social media has become more common in the social sciences (Conrad et al., 2018; Ye et al., 2018; Kärkimaa et al., 2018; Matthews et al., 2018). Statistical or machine learning algorithms[1] are generally gaining in popularity in the social sciences (Oberski, 2018). In a typical supervised learning framework for classifying text answers, a small proportion of texts are coded manually and a statistical learning model is trained on them. The rest of the texts are then coded automatically using the trained model.

The quality of automatic coding depends on the trained statistical model, and the model relies heavily on manually coded data on which it is trained. Unfortunately, human coders can make mistakes due to human error but also because ambiguous texts are difficult to code. To learn about the degree of coding disagreement a common practice is to double code: each text is coded by two human coders and each of them codes without reference to the other (Elias, 1997).

Statistical learning models assume that the training data are coded correctly. A learning model of training data with coding error is likely to perform worse than one without coding error. However, it is unknown whether and how a statistical learning could benefit from double-coded data. In our study, four strategies of double-coding

are proposed, and we compare double-coding and single-coding with respect to their ability to improve automatic coding.

The outline is as follows: Section 2 introduces the application of statistical learning in the context of text classification and relevant studies on inter-coder disagreement. Section 3 proposes four strategies for resolving inter-coder disagreement in double-coding. Section 4 investigates which strategy leads to the highest classification accuracy for a fixed coding budget and where texts are not yet coded. Section 5 investigates which strategy leads to the highest classification accuracy when texts are already double coded; i.e. it considers the question of how to best learn from double coded data. The comparisons in Sections 4 and 5 are based on two data sets of open-ended responses. Section 6 investigates the sensitivity of the results with respect to the cost of an expert coder in one of the double-coding strategies. Section 7 discusses the implications and limitations of the study.

2 Background

2.1 Statistical Learning in Text Classification

Applying statistical learning algorithms has been shown to be effective in classifying answers to open-ended questions. For example, Grimmer and Stewart (2013) surveyed various automatic text coding methods and discussed promises and pitfalls. Joachims (2001) developed a text classification model based on Support Vector Machine (SVM) and achieved good classification performance. Schonlau and

Couper (2016) proposed a semi-automatic algorithm where text answers are coded automatically by multinomial gradient boosting when the probability of correct classification is high and manually otherwise. Gweon et al. (2017) proposed three automatic coding algorithms for occupation coding and showed they improved coding accuracy. King et al. (2017) proposed a computer-assisted statistical method that discovers keywords from unstructured text.

To assess the performance of a statistical learning algorithm on a set of texts, it has to be evaluated on a test data set that is separate from the data the algorithm is trained on. The whole dataset is randomly divided into a training set and a test set. The statistical learning algorithm is trained on the training set, and the predictions of labels on the test set are made based on the fitted model. The algorithm is then evaluated by comparing the predicted classes with the true labels of the texts in the test data (Lewis and Ringuette, 1994; Bijalwan et al., 2014). Cross-validation is an extension of the idea of dividing the data into training and test data.

2.2 Whether and How to Inter-Coder Disagreement Affect Statistical Learning is Unknown

In practice, inter-coder disagreement is common when a text is coded by two or more coders (Ames et al, 2005; Carley, 1993; Crittenden and Hill, 1971; Popping and Roberts, 2009; Schonlau, 2015). Inter-coder agreement is typically measured by Cohen's kappa coefficient, a measure that takes into account chance agreement

among the human coder (Fleiss et al., 2003). To reduce inter-coder disagreement, an iterative process of coding that consists of assessing inter-coder reliability and modifying the codebook is preferred (Hruschka et al., 2004). Any remaining disagreement can be resolved in one of several ways, including: 1) The two coders discuss disagreement and reach a consensus (e.g. D'Orazio et al., 2016). 2) An expert with more experience determines the code. 3) A third coder is employed. The third coder breaks the tie among the first two coders and the code corresponding to the "majority vote" is assigned.

Inter-coder agreement is often used as a diagnostic tool of the reliability of the coding procedure. Typically, the value of kappa is stated as well as whether the value of kappa is large enough to be deemed acceptable. In large data sets often only a subset of the data is double-coded to determine kappa; the remainder is single-coded. Inter-coder reliability strictly refers to the double coded texts; it does not change anything for the single-coded texts or inform the coding of uncoded texts.

In statistical learning the remaining uncoded texts are coded by an algorithm. It is unclear whether to train the algorithm on the codes of both human coders, or on code after resolving coding differences, or something in between. Training the algorithm on both codes better reflects the measurement error represented by the two differing codes. One might also argue that the learning algorithm should be trained on a data set that contains as few errors as possible. This would argue for training only on texts where both coders selected the same code. Finally, what should be the role of experts in resolving inter-coder disagreement? While experts may code with greater accuracy,

they are also more expensive which means we can code fewer texts for training. When the coding budget is fixed, it is unclear how greater accuracy trades off with a smaller training data set.

3 Strategies for Resolving Inter-Coder Disagreement in Double-coding

We propose and evaluate four strategies to resolve any inter-coder disagreement in double-coding:

- Replicate: Replicate double-coded text observations in the training data, once for each coding, regardless of whether the codes are the same or not.
- Remove differences: Remove text observations coded differently by the two coders from the training data.
- Majority vote: Ask a third coder to code only text observations coded differently by the two coders. The code is determined by majority vote.
- Expert resolves: Ask an expert to code text observations coded differently by the two coders. Labels are determined by the expert. It is assumed in the study that the expert is always correct (although not literally true, this assumption approximates experts having higher coding accuracy).

The number of texts in the training data varies depending on the strategy applied. Specifically, the number of observations in the training data is doubled in "replicate"

and reduced in “remove differences”. The number of observations does not change for “majority vote” and “expert resolves”.

Moreover, different strategies lead to different costs for coding an observation. Double-coding costs twice as much as single-coding since two human coders are needed to code one text observation. “Expert resolves” is the most expensive strategy as hiring an expert usually costs much more than hiring a regular coder. The strategies “replicate” and “remove differences” are the least expensive strategies since duplicating or removing an instance requires no additional coding.

The proposed strategies and single-coding are compared in two scenarios: 1) the total cost or budget is fixed but the number of texts in the training data varies depending on the strategy chosen and 2) the data are already double coded; and we can choose among the strategies irrespective of cost.

4 Double Coding Strategies When the Budget is Fixed

In practice, researchers want to control the cost of manual coding given the budget is usually fixed. Therefore, we compare the predictions of the four double-coding strategies and single-coding under a fixed budget. To explore how different coding strategies perform in this case, we conducted experiments based on two datasets: the Patient Joe and the Smokers’ Helpline.

4.1 Data

The Patient Joe dataset contains 1758 responses to an open-ended question in the Internet survey panel LISS (http://www.lissdata.nl). The question concerns the following hypothetical scenario: “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?” This question was intended to study patient activation and medical decision making (Martin et al., 2011). Text answers were coded into one of four categories based on a coding manual (Schonlau and Couper, 2016). Here, we only consider classification into “proactive” vs. “not proactive”. In our experiment, the 1758 observations were randomly divided into a training set (1000 observations) and a test set (758 observations).

The Smokers’ Helpline dataset comes from University of Waterloo Smokers’ Helpline (http://www.smokershelpline.ca), a helpline for Canadian smokers who want to quit smoking. Six months after the initial call there is a follow-up phone survey during which the following open-ended question is asked “What helped you the most in trying to quit (smoking)?”. Responses were recorded and manually coded into one of 27 categories. For our experiment, we consider a binary classification on whether the willpower helped respondents the most in trying to quit smoking. Among the 3352 observations in the dataset, 2000 of them were used for training while the rest of 1352 observations were for testing.

For both data sets, we simulated regular coders' code from the correct classification by randomly changing the correct code to the incorrect code with probability p, where p is the error rate of the simulated coders. That is, for each observation we had the actual code from the data set, a simulated coder's code for single-coding and two independent simulated coders' code for double-coding. Single-coding and the proposed double-coding strategies were applied on the simulated codes.

4.2 Experimental Setup

After applying each of the proposed double-coding strategies on the simulated training data, a statistical learning algorithm, support vector machines (Joachims, 1998) with linear kernel and parameter C=100, was trained. As usual, the value for the tuning parameter was determined through a grid search (C=0.1, 1, 10, 100 and 1000) with C=100 achieving the prediction accuracy on the test data . The predictions of trained models on the test set were evaluated under prediction accuracy. Accuracy measures the percentage of test observations that are correctly coded using the trained model.

As we have pointed out in Section 3, different strategies for double-coding result in different cost to code a text. Also, the cost per observation in "majority vote" and "expert resolves" depends on the coding error rate. For example, with a higher error rate, the two coders would be more likely to code differently, and more work need to

be done by a third coder or an expert. Hence the probability of requiring a third coder or an expert varies with the coding error rate, and so does the cost. Therefore, when the budget in expectation is fixed, the number of coded observations that we can afford depends on the coding error rate and the strategy we apply.

The following strategies all require an expected number of N annotations by a regular coder.

1. Single-code N instances.
2. Double-code $N/2$ instances using strategy “replicate”.
3. Double-code $N/2$ instances using strategy “remove differences”.
4. Double-code $\frac{N/2}{1+p-p^2}$ instances using strategy “majority vote”.
5. Double-code $\frac{N/2}{1+tp-tp^2}$ instances using strategy “expert resolves”, where t is the relative cost of coding by an expert over coding by a regular coder.

The formulas for the strategies "majority vote” and "expert resolves” involve the probability of coding error *p*. Here the number of annotations is not deterministic but is N in expectation (on average). The derivation of these two formulas (strategies 4 and 5) is given in Appendix A. Also, “expert resolves” involves an expert coder. To compare the cost of this strategy with the other strategies, the relative cost of an expert coder is converted to that of a regular coder using the relative cost t. For example, if an expert’s code is twice as expensive as a regular coder, t=2. In our

experiments, we assumed t = 10, i.e., the cost of coding by an expert is ten times as that of coding by a regular coder.

We either single-coded the whole training set or double-coded a random subset of the training set (the subset size can be calculated using the formulas in the earlier part of this section). We assumed the expected budget allows us to single-code the whole training set, i.e., 1000 annotations in the Patient Joe data, and 2000 annotations in the Smokers' Helpline data.

4.3 Experimental Results

Figure 1 shows the averaged prediction accuracy for the five strategies -- single-coding and the four double-coding strategies -- as a function of coding error rate for both datasets. The prediction accuracy is averaged over 100 experiments. The plot suggests that, when the expected coding budget is fixed: 1) As the coding error rate increases, all strategies predict worse; 2) Single-coding outperforms double-coding when the error rate is small ; 3) "Expert resolves" works best when the coding error rate exceeds a data-dependent threshold, which is around 30% in the Patient Joe and 25% in the Smokers' Helpline; 4) When the error rate is close to 50%, "expert resolves" still gets an informative training set, while single-coding and other double-coding strategies become similar to random guessing.

<Figure 1 about here>

5 Learning Strategies When Texts are Already Double Coded

The previous section deals with the scenario that researchers have limited funding and seek a proper strategy under budget to code a training set. Yet in some cases, the day may already have been double coded. Then researchers have to decide how to make use of the data to train a statistical learning model.

Analogous to Section 4, using the Patient Joe and the Smokers' Helpline datasets, we simulated regular coders by changing the correct class to incorrect with probability p (coding error rate).

Figure 2 shows the averaged evaluation of predictions from fitted models in 100 repeated experiments on the Patient Joe and the Smokers' Helpline, as coding error rate changes from 0% to 50%. Because there are two classes, an error rate of 50% corresponds to random guessing and an error rate over 50% means coding is worse than random guessing. The plots show: 1) As the coding error rate increases, prediction accuracy decreases for both single-coding and double-coding; 2) Double-coding improves predictions, especially when the coding error gets large but remains below random guessing (50% error); 3) "Expert resolves" results in better predictions than single-coding and other double-coding strategies, regardless of whether the coding error rate is high or low. Even when the error rate approaches 50%, "expert resolves" still gets an informative training set, while other strategies become similar to random guessing; 4) "Remove differences" is the second best

double-coding strategy and works better than “majority vote”. 5) “Replicate” performs worst among the four double-coding strategies.

<Figure 2 about here>

In conclusion, if text data are already double coded and an expert has resolved the differences, then the expert-resolved data should be used as training data. If no expert was available to resolve differences, it is better remove differences from the training data than to engage a third coder and employ the “majority vote” strategy.

6 Sensitivity Analysis of the Relative Cost of Experts

The conclusion in Section 4 is based on the assumption that the cost for an expert to code a text is 10 times that of a regular coder. If the cost of an expert is lower than assumed, “expert resolves” becomes more cost-efficient and the “expert resolves” strategy becomes preferable for lower error rates. In this section, we analyze how the relative cost of coding by an expert changes the threshold error rate at which “expert resolves” starts to predict most accurately. We investigated the relationship by performing simulated experiments as a function of the relative cost t.

Figure 3 shows -- under fixed budget -- how the threshold error rate changes as the relative cost of coding by experts increases from 1 to 20. As expected, as the relative cost of an expert increases, the threshold at which “expert resolves" beats single coding also increases. However, even when the relative cost of an expert is

extremely high (e.g. 15), “expert resolves" still beats single coding when coders make many mistakes (e.g. error rate > 25~35%).

<Figure 3 about here>

7 Discussion

We have explored whether and how double-coding can be used to improve automatic classification of responses to open-ended questions. Four strategies are proposed for resolving potential inter-coder disagreement in double-coding. We compare these strategies with single-coding in two scenarios: 1) When the budget for manual coding is fixed, single-coding outperforms double-coding when the coding error rate is lower than a data-dependent threshold, while double-coding works better than single-coding otherwise. The threshold error rate is around 20~35% in our experiments. Further, when double-coding is preferable "expert resolves” and “remove differences” are the best and the second-best strategies. 2) When texts have already been double coded, we find letting experts resolve inter-coder disagreement leads to the highest classification accuracy. If an expert is not available, the second-best strategy is to “remove differences” from the training data. “Majority vote” also improves the prediction of automatic coding but not as much.

It is somewhat surprising that removing inter-coder disagreement beats the “majority vote” strategy that uses a third coder. Removing texts with disagreement represents a trade-off: Eliminating most coding errors in exchange for reducing the

size of the training data. A small percentage of coding errors remain as both coders may have miscoded with a probability of $(1-p)^2$. We conclude that you would rather have a cleaner than a large data set. Of course, when generalizing to more than two outcome classes, it is not clear whether "remove differences" would still beat "majority vote".

Coding errors can be due to human error or due to ambiguity of the text. (We include incomplete coding manuals in the category of human errors, even though it is not the fault of the coder.) A text may be ambiguous because it contains contradictory information, not enough information, or information unrelated to the question. Human errors can be reduced by using multiple coders or an expert. Truly ambiguous texts cannot be classified and sending an ambiguous text to an expert would not be helpful. The experiments have focused on the human error. In practice, ambiguous texts should probably be removed, but the boundary between ambiguous and human error may not always be clear.

Rather than choosing training data at random, the goal of active learning is to purposefully select the training data in hopes of either needing fewer training observations for the same performance or improving the performance for a fixed number of training observations. Tong and Koller (2001) showed that active learning in text classification can significantly reduce the number of training instances without deteriorating performance metrics. Incorporating active learning to select training data for double-coding may improve the performance of automatic coding.

The limitations of this study include: 1) Although we identify the existence of a threshold between single-coding and double-coding, we do not know what its value is for a specific dataset. The data-dependent threshold in our experiments was between 20% to 35%. This suggests that researchers may use single-coding if they think regular coders has coding accuracy over 80% while apply "expert resolves" double-coding strategy if coding accuracy is less than 65%. 2) In the experiments, we only use SVM to train a statistical learning model. While SVM is one of the most commonly used methods in text classification, other modern statistical learning algorithms (random forest, gradient boosting, etc.) could be used also. 3) We only consider strategies for binary classification problems, and more work needs to be done to extend to multi-class cases. For example, in multi-class cases, "majority vote" may not be able to solve inter-coder disagreement if the third coder codes differently from the first two. It may be harder to code correctly when there are many classes to choose from. We conjecture that increased coding error may more often lead to the decision to double code as compared to binary classification. 4) In the experiments, we assume the coding error rate is constant across all coders and observations. This implies different coders are equally likely to make a mistake and that mistakes are equally likely for any coding any text. While this is unlikely to be true, we believe it is probably a reasonable approximation. 5) We assumed the coding mistakes made by the simulated coders are independent in our experiments. However, the coding errors of the same coder may be correlated as a coder is likely to make the same mistakes repeatedly. So our experiments may even underestimate the performance of

double-coding, because a second opinion may counterbalance the bias of a particular coder.

Although the assumption that the expert is always correct is not literally true in practice, we believe it is a reasonable approximation. The expert does code with much greater accuracy than a regular coder. In practice, we would not know the coding error of an expert and assuming zero error facilitates the simulation. If we allowed the expert to have a modest coding error in the simulation, the results would be qualitatively the same; of course the threshold would shift somewhat. To verify this, we reran the experiments assuming the expert's coding error rate was one tenth of that of a regular coder (not shown). The threshold at which the strategy "expert resolves" is preferable over single coding became slightly larger (by about 2%), meaning that single coding remains attractive at slightly higher coding inaccuracies. Otherwise, results were consistent with previous results.

What should you do if only a subset of the data is double coded while the remainder is single-coded? While we have not formally investigated this scenario, our results suggest using the "expert resolves" or "remove differences" strategies on the double coded data and combining this with the single coded data. If only a small subset is double coded, we would expect only a marginal overall improvement on accuracy.

This paper is focused on the coding procedure in classifying short texts such as open-ended responses. D'Orazio et al. (2016) have taken a different approach to reduce the cost of manually coding all texts. Rather than employing statistical

learning, they recruited coders on the Amazon's crowdsourcing platform "Mechanical Turk" who are generally paid much less than regular coders. Recruiting coders on "Mechanical Turk" may be advantageous under some circumstances: 1) when sample sizes are small or moderate (the statistical learning approach still requires the same training data whether the sample size is large or not); 2) when the task is very complex, for example if the task may require a text answer 3) when the text is relatively long because the ngram approach to statistical learning does not tend to work well on long texts.

In summary, when classification is error prone double coding is preferable to single coding. Among double coding strategies, using an expert is preferable even considering the increased cost of the expert. When no expert is available, one should remove differently coded data from the training data – even though this reduces the training data set.

Acknowledgement

This research was supported by the Social Sciences and Humanities Research Council of Canada (SSHRC # 435-2013-0128). We gratefully acknowledge the Canadian Cancer Society Smokers' helpline (https://www.smokershelpline.ca/) for permission to use their data. We are equally happy to make use of "Patient Joe" data of the LISS (Longitudinal Internet Studies for the Social sciences) panel administered by CentERdata (Tilburg University, The Netherlands).

Author Information

Zhoushanyue He is a Ph.D. student in the department of Statistics and Actuarial Science at the University of Waterloo. Her interests include statistical learning and survey methodology. She can be reached at z26he@uwaterloo.ca.

Matthias Schonlau is a Professor of statistics in the department of Statistics and Actuarial Science at the University of Waterloo. His interests include survey methodology and in particular methodology for the analysis of open-ended questions. He can be reached at schonlau@uwaterloo.ca or through his website www.schonlau.net .

Data Availability

The "Patient Joe" data are available from the LISS panel upon request https://www.lissdata.nl/ . The Smokers' Helpline data are available for replication purposes from the second author at schonlau@uwaterloo.ca.

Software Information

The computation in the paper is programmed using R 3.4.1, with package "e1071" to implement SVM models. Code for replication is available from the corresponding author.

References

Ames, S. L., Gallaher, P. E., Sun, P., Pearce, S., Zogg, J. B., Houska, B., Leigh, B. C., and Stacy, A. W. (2005). A web-based program for coding open-ended response protocols. *Behavior Research Methods*, 37(3): 470-479.

Bijalwan, V., Kumar, V., Kumari, P., and Pascual, J. (2014). KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, 7(1): 61-70.

Carley, K. (1993). Coding choices for textual analysis: a comparison of content

analysis and map analysis. *Sociological Methodology*, 23: 75-126.

Conrad F., Gagnon-Barsch J., Ferg R., Hou E., Pasek J., Schober M. (2018). Social media as an alternative to surveys of opinions about the economy. *Paper presented at BigSurv18*, Barcelona, Spain (Under final revision for a special issue of Social Science Computer Review).

Crittenden, K. S., Hill, R. J. (1971). Coding reliability and validity of interview data. *American Sociological Review, 36(6):* 1073-1080.

D'Orazio, V., Kenwick, M., Lane, M., Palmer, G., & Reitter, D. (2016). Crowdsourcing the measurement of interstate conflict. *PLoS ONE*, *11*(6): e0156527.

Elias, P. (1997). Occupational classification (ISCO-88): concepts, methods, reliability, validity and cross-national comparability. *OECD Labour Market and Social Policy Occasional Papers*, 20. Retrieved from https://EconPapers.repec.org/RePEc:oec:elsaaa:20-en

Fleiss, J. L., Levin, B., and Paik, M. C. (2003). Statistical methods for rates and proportions (3rd ed.). New York: John Wiley & Sons.

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.

Gweon, H., Schonlau, M., Kaczmirek, L., Blohm, M., and Steiner, S. (2017). Three

methods for occupation coding based on statistical learning. *Journal of Official Statistics*, 33(1): 101-122.

Hruschka, D. J., Schwartz, D., St. John, D. C., Picone-Decaro, E., Jenkins, R. A., and Carey, J. W. (2004). Reliability in coding open-ended data: lessons learned from HIV behavioral research. *Field Methods*, 16(3): 307-331.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, 137-142. London, UK. Springer.

Joachims, T. (2001). A statistical learning model of text classification for support vector machines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 128-136. New Orleans, USA. ACM.

Kärkimaa J. Larja L. (2018). How to make AI do your job for statistical classification of industry and occupation. *Paper presented at BigSurv18*, Barcelona, Spain.

King, G., Lam, P., and Roberts, M. E. (2017). Computer-assisted keyword and document set discovery from unstructured text. *American Journal of Political Science*, *61*(4): 971-988.

Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 81-93. Las Vegas, USA.

University of Nevada.

Martin, L. T., Schonlau, M., Haas, A., Derose, K. P., Rosenfeld, L., Buka, S. L., and Rudd, R. (2011). Patient activation and advocacy: which literacy skills matter most? *Journal of Health Communication*, 16(sup3): 177-190.

Matthews P, Kyriakopoulos, G., Holcekova M. (2018). Machine learning and verbatim survey responses: classification of criminal offences in the crime survey for England and Wales. *Paper presented at BigSurv18*, Barcelona, Spain.

Oberski D. (2018). Can Facebook "likes" measure human values? *Paper presented at BigSurv18*, Barcelona, Spain (Under review for a special issue of Social Science Computer Review).

Popping, R. and Roberts, C. W. (2009). Coding issues in modality analysis. *Field Methods*, 21(3): 244-264.

Schonlau, M. (2015). What do web survey panel respondents answer when asked "Do you have any other comment?". *Survey Methods: Insights from the Field (SMIF)*. Retrieved from https://surveyinsights.org/?p=6899

Schonlau, M. and Couper, M. P. (2016). Semi-automated categorization of open-ended questions. *Survey Research Methods*, 10(2): 143-152.

Tong, S., and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*: 45-66.

Ye C., Medway R., Kelley C. (2018). Natural language processing for open-ended survey questions. *Paper presented at BigSurv18*, Barcelona, Spain.

Endnotes

1) We use the terms “statistical learning” and “machine learning” as synonyms.

Appendix A: Derivation of the Number of Training Observations under Fixed Budget

The purpose of this appendix is to derive the formulas for the number of training observations for strategies 4 “majority vote” and 5 “expert resolves” in Section 4.2. We begin with the formula for strategy 5 because the formula for strategy 4 can be derived as a special case.

Strategy 5: The fixed budget allows for N annotations by a regular coder. If no expert were required, we could double code N/2 texts. We can only afford a smaller number of annotations, say M, because experts still have to resolve any coding disagreement. An expert is t times as expensive as a regular coder. As each regular coder has probability p to code incorrectly, the probability that two independent coders disagree is 2p(1-p). Therefore, the expected cost for using “expert resolves” to code M observations is

$$2M + 2Mtp(1-p)$$

The first term of the above formula is the cost for having two regular coders to code the M observations, and the second term is the (expected) cost for having an expert to resolve the inter-coder disagreement between the two regular coders.

Under fixed budget equivalent to N annotations by a regular coder, we have

$$2M + 2Mtp(1-p) = N$$

The number of training observations for “expert resolves” under fixed expected

budget of N annotations is obtained by solving for M

$$M = \frac{N}{2 + 2tp(1-p)} = \frac{N/2}{1 + tp - tp^2} \qquad \text{(A.1)}$$

Strategy 4: In deriving the formula for "majority vote", we note that we have a regular coder instead of an expert to solve inter-coder disagreement. Therefore, "majority vote" is a special case of "expert resolves" with t=1. The number of training observations for "majority vote" under fixed expected budget of N annotations follows from (A.1) with t=1:

$$\frac{N/2}{1 + p - p^2}$$

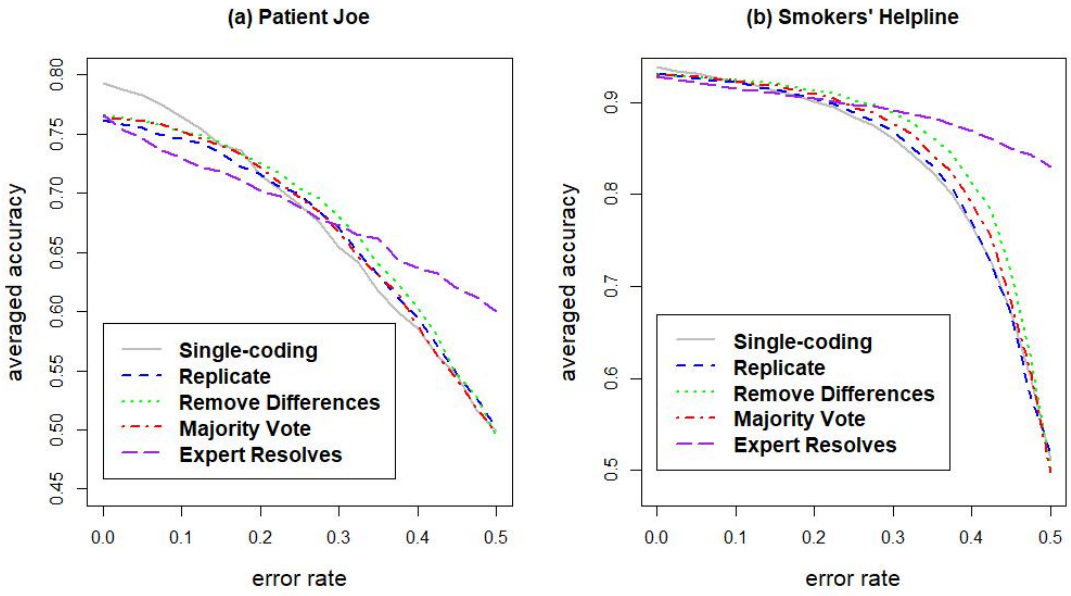

Figure 1: Averaged accuracy of predictions when the expected budget is fixed, with (a)the Patient Joe data and (b)the Smokers' Helpline data. It is assumed that the cost for an expert to code an instance is 10 times that of a regular coder.

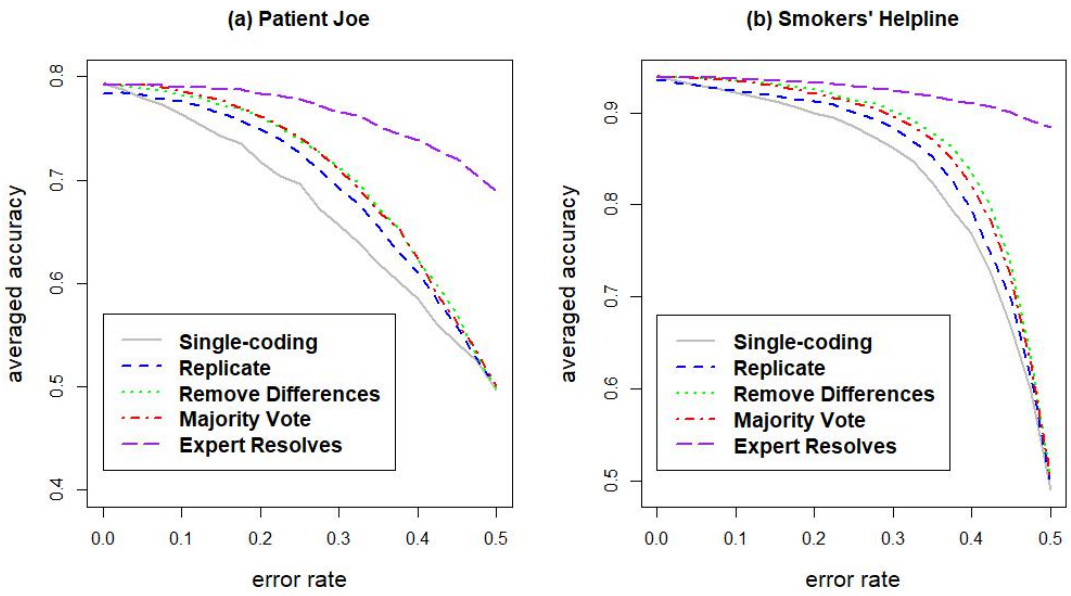

Figure 2: Averaged accuracy of predictions when the texts have already been double coded, with (a)the Patient Joe data and (b)the Smokers' Helpline data.

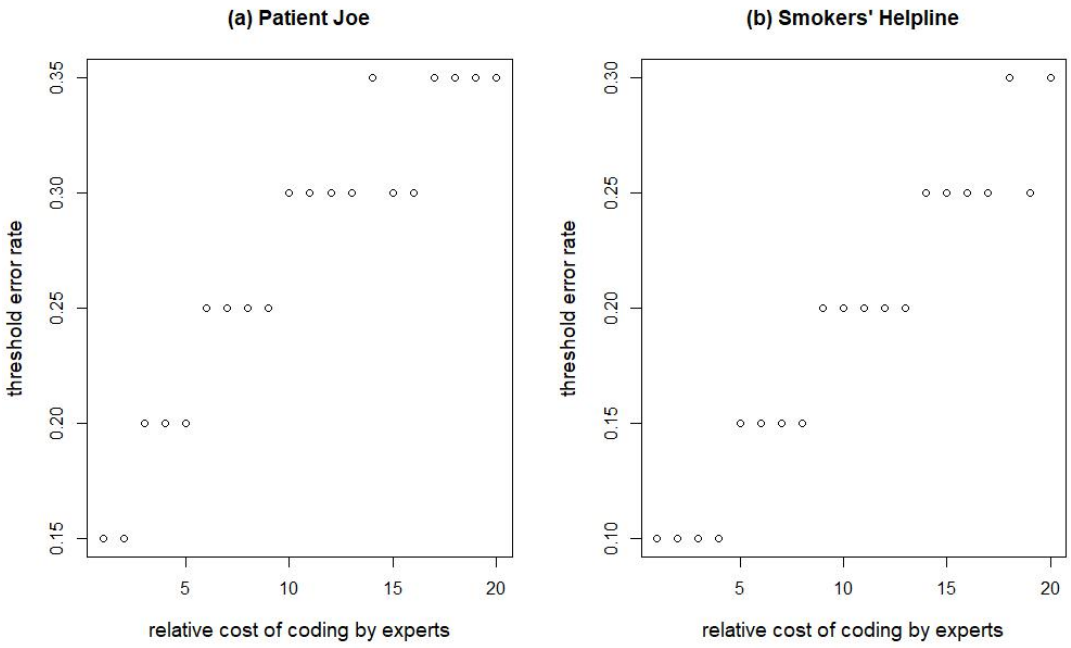

Figure 3: Threshold error rate that "expert resolves" outperforms single-coding vs. the relative cost of coding by an expert. For a specific level of relative cost, if coding error rate is less than the threshold error rate, single-coding results in more accurate predictions than "expert resolves"; if coding error rate is larger than or equal to the threshold error rate, "expert resolves" predicts better than single-coding.