Survey Research Methods (2020)
Vol. 14, No. 3, pp. 267-278
doi:10.18148/srm/2020.v14i3.7639


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

# Automatic Coding of Open-ended Questions into Multiple Classes: Whether and How to Use Double Coded Data

Zhoushanyue He
University of Waterloo
Canada

Matthias Schonlau
University of Waterloo
Canada

Responses to open-ended questions in surveys are usually coded into pre-specified classes, manually or automatically using a statistical learning algorithm. Automatic coding of open-ended responses relies on a set of manually coded responses, based on which a statistical learning model is fitted. In this paper, we investigate whether and how double coding can help improve the automatic classification of open-ended responses. We evaluate four strategies for training the statistical algorithm on double coded data, using experiments on simulated and real data. We find that, when the data are already double coded (i.e. double coding does not incur additional costs), double coding where an expert resolves intercoder disagreement leads to the greatest classification accuracy. However, when we have a fixed budget for manually coding, single coding is preferable if the coding error rate is anticipated to be less than about 35% to 45%.

*Keywords:* Open-ended question; Double coding; Text coding; Text classification; Statistical learning; Machine learning

## 1 Introduction

Open-ended questions allow researchers to ask questions without constraining respondents' answer choices and without accidentally biasing them towards more socially desirable responses. However, open-ended questions yield text responses and text is hard to analyze quantitatively: logistic and linear regression require numerical data. Therefore, text answers are often categorized (or classified) into classes based on a coding manual.

When text responses to open-ended questions need to be categorized, there are two choices: manual coding and automatic coding. Manual coding refers to having a human coder decide in which class a response should be, while automatic coding refers to categorizing responses based on statistical learning models. Manual coding is expensive because it requires human coders. Therefore for large data sets, automatic (Gweon, Schonlau, Kaczmirek, Blohm, & Steiner, 2017; Schierholz, 2019) or semi-automatic coding (Schonlau & Couper, 2016) may be attractive. In automatic coding, a statistical learning model is trained on a smaller set of manually coded data, which is called the training data. The model is then used to predict the code of uncategorized text answers. The advantage of automatic coding is reduced cost and fast speed (relative to human coding). One disadvantage of automatic coding is the need for expertise to execute the modelling and prediction.

Because automatic coding predicts the classes of texts based on a trained model, its performance is influenced by manually coded data the model is trained on. The quality of manually coded data depends on human coders, and ideally coders make no mistake. Unfortunately, in practice, coders do make mistakes due to human errors or the ambiguity of responses. The coding error in the manually coded data deteriorates the performance of automatic coding (Mullainathan & Obermeyer, 2017). How much the performance deteriorates is poorly understood. Given that some coding error is unavoidable, studying whether automatic coding should explicitly address coding error is worthwhile.

To ascertain the extent of intercoder disagreement, researchers often double code a subset of the data using two different coders. When double coded data are available, it is unclear whether the statistical learning model should be trained on the codes of both coders, or on the codes after inter-coder disagreement is resolved, or something in between. Alternatively, if the texts have not yet been coded and the budget for manual coding is fixed, the statistical learning model may still benefit from double coding. There is a tradeoff between a larger single coded training data set, and a smaller, higher quality double coded training data set. We consider this tradeoff using a simulation and apply the proposed methodology to two data sets.

In earlier work, we investigated whether and how statis-

Zhoushanyue He, University of Waterloo, 200 University Ave West, Bldg M3, Waterloo ON, Canada N2L 3G1 (E-mail: z26he@uwaterloo.ca)

tical learning algorithms should use double coding for *binary* classification, i.e. when there are only two possible codes (He & Schonlau, 2019). This applies, for example, to choose-all-that-apply questions where an answer does or does not mention each answer category. We found when the coding budget is fixed, double coding outperforms single coding if the coding error rate exceeds a threshold. When double coded data are already available, asking an expert to resolve inter-coder disagreement is preferable to resolving disagreement by majority vote or by not including texts with disagreement in the training data.

In this paper, we extend the idea of double coding as part of statistical learning from binary classification to multi-class classification. For binary classification, there is only one type of coding mistake: the true class was the other class. For multi-class classification the situation is more complex: simulations of intercoder disagreement require additional assumptions on the structure of the misclassification matrix.

The outline of this paper is as follows: Section 2 reviews relevant literatures on double coding and the application of statistical learning to text classification. Section 3 introduces double coding in the context of multi-class classification. Section 4 presents experiments on simulated codes; such simulated experiments allow adjusting the coding error rate to observe the performance of different coding strategies in various scenarios. Section 5 verifies our finding from simulated experiments by applying the proposed coding strategies on two double coded data sets. Section 6 concludes with a discussion.

## 2 Background

Responses to open-ended questions in surveys are often text data. Statistical learning has been widely used in analyzing text data. For instance, Joachims (2001) used Support Vector Machines (SVM) to develop a model to classify text and showed good generalization performance of the model. Schonlau and Couper (2016) applied multinomial gradient boosting in a semi-automatic algorithm, which coded text answers that were likely to be correctly classified automatically and manually. Kern, Klausch, and Kreuter (2019) discussed previous and prospective applications of tree-based statistical learning methods (random forest, boosting, etc.) such as classifying text answers and modeling nonresponse in survey research.

One example of text data in surveys is occupation coding. It refers to coding the text answer of an open-ended question about one's job. Applying statistical learning algorithms on occupation coding has become increasingly common. Schierholz (2019) compared statistical learning algorithms in occupation coding. Gweon et al. (2017) proposed three automatic coding algorithms and improved coding accuracy for occupation coding.

Another example of text data in surveys is responses to probing questions. Probing questions are follow-up questions asking respondents to provide additional information about a survey item (Beatty & Willis, 2007; Meitinger, Braun, & Behr, 2018). Behr, Kaczmirek, Bandilla, and Braun (2012) classified answers to probing questions into two classes, productive and nonproductive answers, and tested whether an increasing number of preceding probing questions influenced the quality of the answers.

In order to apply statistical learning methods in coding text responses, we have to fit a model on a set of data (training data) and then use the fitted model to predict the codes for some other data (test data). Usually, more training data means the trained algorithm performs better. More classes and more features typically require more training data. There is no strict guidance on the size of training set in the literature. Schierholz (2019) suggested that the training set should be large enough to contain a variety of potential texts (including misspellings) to cover all contingencies how a specific text can be coded into different classes. Moreover, if the training data do not cover some of the categories, these categories would never be suggested by predictions based on the training data only. Learning competitions usually have large training data sets (with known responses). Here, the text answers for training have to be manually coded first, which is costly. To avoid large costs, we need to balance our desire to predict well – requiring a large training data set – with our desire to keep the costs down – requiring a small training data set. Schonlau and Couper (2016) have used a training data set of size 500 for four outcome classes.

Statistical learning on text responses requires training data which are manually coded. For manual coding, there are usually multiple coders either to speed up the coding process or to compute intercoder reliability, which enhances objectivity and quality (Ames et al., 2005; Carley, 1993; Popping & Roberts, 2009; Schonlau, 2015). It is natural that different coders have different opinions on some texts (Conrad, Couper, & Sakshaug, 2016), which may due to ambiguity of texts, lack of clarity of the coding manual or different personal understanding.

The extent of inter-coder agreement is usually measured by Cohen's kappa coefficient (Fleiss, Levin, & Paik, 2013). Inter-coder agreement for classifying open-ended questions is often low. For example, Elias (1997) and Mannetje and Kromhout (2003) found that the agreement rates for occupation coding are 55% to 80% at 3-digit level. Researchers have considered assessing inter-coder reliability and modifying codebook as an iterative process to reduce inter-coder disagreement (Hruschka et al., 2004). Remaining coding disagreements can be resolved through 1) a discussion of the two coders until a consensus is reached (D'Orazio, Kenwick, Lane, Palmer, & Reitter, 2016), 2) adding a third coder and deciding by majority vote, or 3) letting an expert decide.

## 3 Methodology

In binary classification, as a response can only be in one of the two classes (assumed to be class A and class B), a coder can only make two coding mistakes: coding a response from A incorrectly to B and coding a response from B incorrectly to A.

In multi-class classification, the number of misclassification errors is larger. We use a coding matrix to represent the coding performance of a regular coder. The coding matrix is a $L * L$ matrix, where $L$ is the number of classes. The $(i, j)^{th}$ element of the coding matrix $p_{ij}$ represents the probability that a regular coder codes a text corresponding to class $i$ into class $j$. The coding matrix can then be written as

$$M = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1L} \\ p_{21} & p_{22} & \cdots & p_{2L} \\ \vdots & \vdots & \vdots & \vdots \\ p_{L1} & p_{L2} & \cdots & p_{LL} \end{pmatrix} ,$$

where $\sum_{j=1}^{L} p_{ij} = 1$. The coding matrix for binary classification in He and Schonlau (2019) is a special case:

$$M_{binary} = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} ,$$

where $p$ is the error rate. When two coders classify a response independently, they may assign different codes. We evaluate the following strategies to deal with the inter-coder disagreement:

- Single coding: each text is coded by a regular coder into one of the classes.
- Replicate: replicate each double coded text into two texts, one with each of the double codes, no matter whether the double codes are the same or not.
- Remove differences: texts that are coded differently by the two coders are removed from the data.
- Majority vote: if a text is coded differently by the two coders, a third coder codes. For simplicity, we assume the third coder can only choose a code from the first two codes. Thus, the third code leads to a 2:1 majority.
- Expert resolves: an expert coder arbitrates any inter-coder disagreement.

When responses have already been double coded, we can apply any of the above strategies (for single coding one must choose one of the two coders' code). When texts are not yet coded and the budget for manual coding is fixed, the cost of applying "replicate" or "remove differences" is twice that of single coding, while the cost of "majority vote" or "expert resolves" is more than twice that of single coding. Therefore, the number of responses we can afford to code under a fixed budget using different coding strategies varies. The number of texts we can afford to code under a fixed budget for "replicate" and "remove differences" is half of that of single coding as we spend two annotations on each text. If we denote the number of texts for "single coding" as $N$, then for "replicate" and "remove differences", the number under a fixed budget is

$$N/2 \tag{1}$$

For "majority vote", the number of texts under a fixed budget is

$$N/(3 - \sum_{i=1}^{L} q_i \sum_{j=1}^{L} p_{ij}^2) \tag{2}$$

and that for "expert resolves" is

$$N/(2 + t - t \sum_{i=1}^{L} q_i \sum_{j=1}^{L} p_{ij}^2) \quad , \tag{3}$$

where $t$ is the relative cost of coding by an expert vs. coding by a regular coder, and $q_1$, $q_2$, ..., $q_L$ are the marginal distribution of the classes. The derivation of the formulas can be found in Appendix A.

We use the accuracy of automatic coding as the evaluation criterion for comparing the strategies. Accuracy is defined as the fraction of correctly coded observations, i.e. the text responses for which the predicted class matches the true class.

The general coding matrix $M$ contains $L(L-1)$ parameters. In practice, the coding matrix is unknown and contains too many parameters to estimate. Therefore, we consider three special coding matrices: one with equal misclassification probabilities, one with misclassification in neighboring classes, and one with misclassification in higher classes. The coding matrix with equal misclassification probabilities represents the case where coding error happens at random with equal probabilities. It may be a good default choice. The other two coding matrices we consider, one with misclassification in neighboring classes and one with misclassification in higher classes, represent two specific coding error structures: in the first case coders miscode only into neighboring classes, and in the second case have a tendency to code a higher class. We choose these coding matrices because they are, in our opinion, the simplest choices. More complex coding matrices exist, of course. For a specific data set, researchers may decide which special case fits the problem at hand.

### 3.1 Coding Matrix 1: Equal Misclassification Probabilities

A coder has probability $1 - p$ to code a text correctly and probability $p/(L-1)$ to code it into any of the incorrect

classes. The coding matrix is as follows:

$$M_1 = \begin{pmatrix} 1-p & p/(L-1) & \dots & p/(L-1) \\ p/(L-1) & 1-p & \dots & p/(L-1) \\ \vdots & \vdots & \vdots & \vdots \\ p/(L-1) & p/(L-1) & \dots & 1-p \end{pmatrix}.$$

In other words, a coder has coding error rate $p$, and, he/she is equally likely to classify a response into any incorrect class if a mistake happens.

Using formulas 1, 2 and 3, assuming the coding matrix is $M_1$, the number of texts that can be coded under a fixed budget of N annotations is in Table 1. Table 1 also contains special cases for specific values of the error rate $p$. Unlike the general formulas 2 and 3 which are derived in Appendix A, the formulas in Table 1 do not depend on the marginal class distribution $\{q_i\}$.

### 3.2 Coding Matrix 2: Misclassification in Neighboring Classes

Some classes are naturally ordered. For example, in the Patient Joe data that are introduced in Section 4, we classify text answers into four ordered classes: proactive, somewhat proactive, passive and destructive. This second coding matrix is appropriate for ordered classes:

$$M_2 = \begin{pmatrix} 1-p & p & 0 & 0 & \dots & 0 & 0 \\ p/2 & 1-p & p/2 & 0 & \dots & 0 & 0 \\ 0 & p/2 & 1-p & p/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & p/2 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1-p & p/2 \\ 0 & 0 & 0 & 0 & \dots & p & 1-p \end{pmatrix}$$

The matrix suggests that a coder has probability of $p$ to incorrectly classify a response into a neighboring class, and if there are two neighboring classes, the probability of classifying into any of them is equal (i.e. $p/2$).

### 3.3 Coding Matrix 3: Misclassification in Higher Classes

The third special case we consider is also for ordered classes. It assumes the coding matrix of a regular coder is:

$$M_3 = \begin{pmatrix} 1-p & p(1-g_1) & \dots & p\prod_{i=1}^{L-3} g_i(1-g_{L-2}) & p\prod_{i=1}^{L-2} g_i \\ 0 & 1-p & \dots & p\prod_{i=1}^{L-4} g_i(1-g_{L-3}) & p\prod_{i=1}^{L-3} g_i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & p(1-g_1) & pg_1 \\ 0 & 0 & \dots & 1-p & p \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

This coding matrix represents a coder who has a personal tendency to code responses into "higher" classes. The parameters $g_1, g_2, \dots, g_{L-2}$ show the strength of personal tendency. An example of personal tendency is that an optimistic coder may consider responses to be in more "optimistic" classes.

## 4 Experiments on Simulated Data

To explore which coding strategy to use in the three special coding matrices proposed in Section 3, we run experiments based on the Patient Joe data set (Schonlau, 2020). The Patient Joe data set contains 1758 answers to the following open-ended question: "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). This question was used to investigate patients' decision making. The study was fielded in Dutch in the LISS panel[1] in 2012. The responses in this data set have been classified by two coders into one of four ordered classes: proactive, somewhat proactive, passive and destructive. The differences between them were resolved by an expert, which yielded the "gold standard" classification (Schonlau & Couper, 2016).

We converted the text data into unigram and bigram variables (Schonlau, Guenther, & Sucholutsky, 2017). A unigram variable counts occurrence of individual words and a bigram variable counts occurrence of two-word sequences. Because the number of unique words across all texts is rather large, this approach creates a large number of unigram and bigram variables. Benefits of using unigram and bigram variables (or more generally, n-gram variables) include simplicity and scalability. We used stemming in Dutch and removed stopwords. Unigram and bigram variables that did not appear in at least 5 texts were removed. We randomly chose 1000 responses for training and the remainder for testing. Under a fixed budget, the size of training set is calculated using formulas 1, 2 and 3 with $N = 1000$.

For the training set, we simulated coding errors by changing the "gold standard" class to another class based on probabilities specified in the coding matrix. Also, we assume that experts are 10 times as expensive as regular coders (t=10). Automatic coding requires choosing a statistical learning model. We fit SVM models with a linear kernel, a common choice for text data (Joachims, 2001). The value of the tuning parameter C was set to 100 based on an experiment with the manually coded data.

In "majority vote", the third coder is only supposed to choose between the two codes already chosen by the first two coders. We assume that the probability that the third coder chooses between the first two codes is proportional to the probability in the coding matrix. For example, assuming the coding matrix is $M_2$, if a text in Class 2 is coded into Class 1 and 2 by the first two coders respectively, the third coder

[1] http://www.lissdata.nl

Table 1
*Number of texts coded under a fixed budget of N annotations when the coding matrix is $M_1$.*

| Strategy | Number of texts coded under fixed budget | When $p = 0.1$ | When $p = 0.2$ |
|---|---|---|---|
| Single coding | $N$ | $N$ | $N$ |
| Replicate | $N/2$ | $N/2$ | $N/2$ |
| Remove difference | $N/2$ | $N/2$ | $N/2$ |
| Majority vote | $\frac{N}{2+2p-p^2L/(L-1)}$ | $\frac{N}{2.2-0.01L/(L-1)}$ | $\frac{N}{2.4-0.04L/(L-1)}$ |
| Expert resolves | $\frac{N}{2+2tp-tp^2L/(L-1)}$ | $\frac{N}{2+0.2t-0.01tL/(L-1)}$ | $\frac{N}{2+0.4t-0.04tL/(L-1)}$ |

has probability $p/(2-p)$ to choose Class 1 and probability $2(1-p)/(2-p)$ to choose Class 2.

### 4.1 Coding Matrix with Equal Misclassification Probabilities

Using the coding matrix $M_1$, we run experiments on the Patient Joe data with simulated coding. Figures 1a and 1b show the average predictive accuracy as a function of the error rate $p$ for various strategies. For each value of $p$ the experiment was repeated 100 times.

When the double coded texts are already available (Figure 1a), "expert resolves" is the best strategy to resolve inter-coder disagreement, followed by "remove differences". Note that "single coding" and "majority vote" perform similarly. When the budget is fixed (Figure 1b), no single strategy dominates: for low error rates single coding is best, for high error rates "expert resolves" is best. The threshold for the transition is about 35%.

### 4.2 Coding Matrix with Misclassification in Neighboring Classes

Assuming the coding matrix is $M_2$, we run experiments on the Patient Joe data with simulated coding. Figures 1c and 1d show the predictive accuracy as a function of the error rate $p$ averaged over 100 repeated experiments. Unlike for coding matrix $M_1$, we have to assume a marginal distribution of the classes for the simulation (There was no need to do so for $M_1$ because the results in Table 1 did not depend on the marginal distribution $q_i$). We assume the marginal distribution of classes is distribution 1 in Table 2.

In Figure 1c, we observe a similar pattern as we have seen for coding matrix $M_1$. "Expert resolves" is the best strategy when double coded texts are already available. In Figure 1d, under a fixed budget, single coding works better than double coding strategies for small and moderate error rates $p$, and "expert resolves" is best when $p$ gets large.

### 4.3 Coding Matrix with Misclassification in Higher Classes

Assuming the coding matrix is $M_3$, we run 100 repeated experiments on the Patient Joe data with simulated coding. The average predictive accuracy as a function of the error rate $p$ is shown in Figures 1e and 1f. We also assume the marginal distribution of classes is distribution 1 in Table 2.

The parameters $g_i$ are simulation parameters that represent the tendency of a coder to consider a response in a "higher" class. In the Patient Joe data, $L = 4$. We assume here that $g_1 = 0.2$ and $g_2 = 0.2$. Such an assumption suggests that coders have a mild tendency to misclassify into higher classes, and if they make such a mistake, about 80% of times the misclassification will result in the neighboring higher class. The experiment results with other combinations of $g_1$ and $g_2$ are in Appendix B. The results are similar.

For coding matrix $M_3$, we find that "expert resolves" improves prediction most when double coded texts have already been available. Under a fixed budget, for small error rates single coding works better, and for large error rates "expert resolves" outperforms others. Based on the experiments, single coding works better than double coding, unless the error rate is large ($> 45\%$). "Remove differences" is no longer the second-best double coding strategy as computed for $M_1$ and $M_2$. Instead, "majority vote" is the second best when double coded texts have already been available, followed by "replicate".

### 4.4 Robustness of the Marginal Class Distribution

Because coding matrices $M_2$ and $M_3$ depend on the marginal class distributions and using incorrect class distribution may lead to inaccurate estimation of the number of texts that can be coded under a fixed budget, we now investigate the sensitivity of the results using different class distributions. Specifically, we assume the classes are almost uniformly distributed (distribution 2 in Table 2).

Figure 2 shows the results: When double coded texts are available, the class distribution has no effect. When the budget is fixed, although the basic pattern of the performance

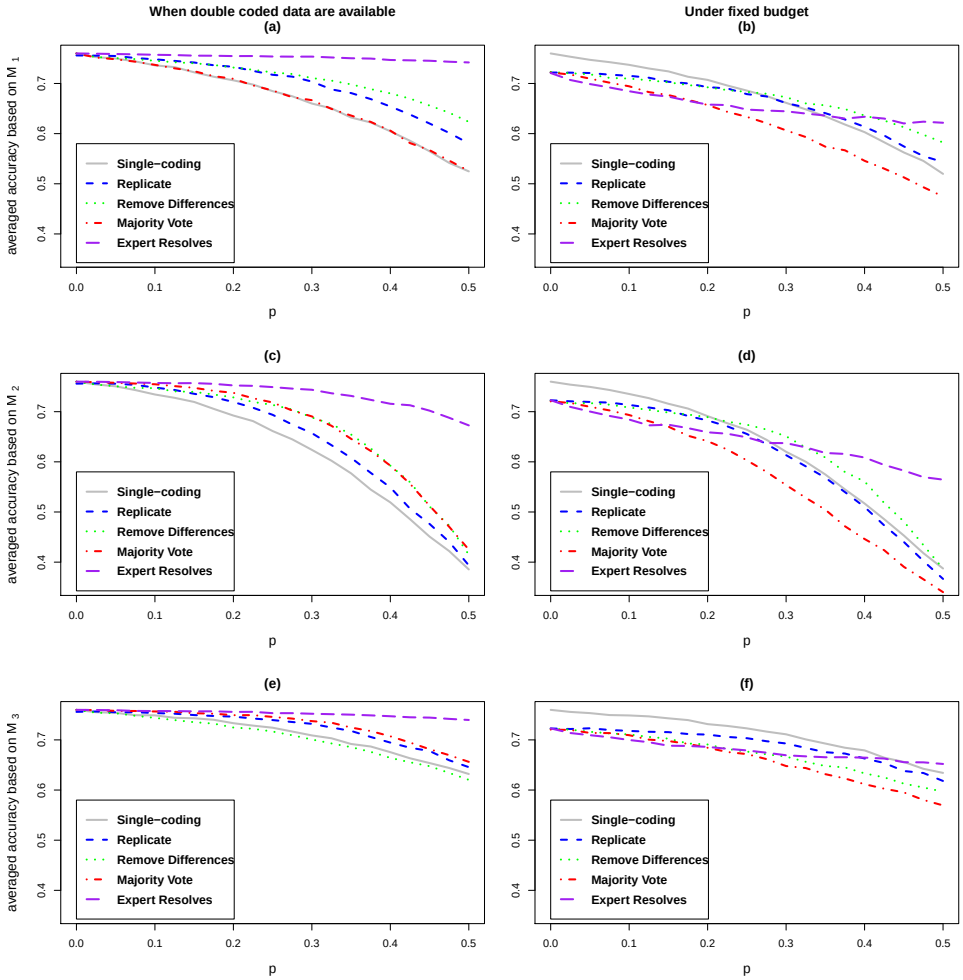


*Figure 1*. Averaged accuracy as a function of error rate $p$ in simulated experiments using the Patient Joe data. Each row represents a different coding matrix ($M_1$, $M_2$ and $M_3$). The coding matrix $M_3$ has parameters $g_1 = 0.2$ and $g_2 = 0.2$. The first column shows the results when double coded data are available, while the second column shows the results when the budget is fixed.

Table 2
*Assumed class distributions for the Patient Joe data*

| Distribution Type | Proactive | Somewhat Proactive | Passive | Destructive |
|---|---|---|---|---|
| Distribution 1 | 0.1 | 0.3 | 0.1 | 0.5 |
| Distribution 2 | 0.3 | 0.3 | 0.2 | 0.2 |

curves is the same, using a more uniform distribution of classes increases the threshold between single coding and "expert resolves". This probably has not much impact in practice: if the coding error is large, the coding procedure should be redesigned.

## 5 Two Case Studies of Applying Double Coding Strategies to Data

In Section 4, we simulated the double codes assuming coders follow the coding matrix. In practice, coding errors do not exactly correspond to a specific coding matrix. The results need to be robust to mild violations of the coding matrix assumption. Therefore, we apply the strategies on two

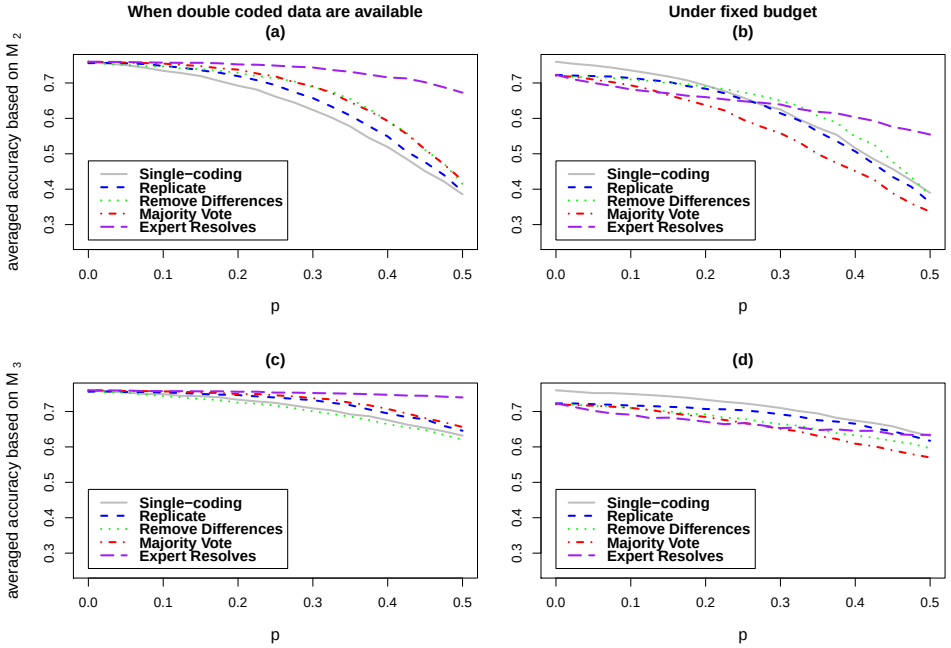


*Figure 2*. Sensitivity analysis for the Patient Joe data with different marginal class distributions. Otherwise it is analogous to Figure 1.

double coded data sets: Happiness and Patient Joe. In both the Patient Joe and the Happiness data sets, two coders coded all data independently, and the disagreement between them was resolved by experts. We can implement all strategies on the two data sets based on available codes (except "majority vote" due to the lack of a third coder).

The Happiness data 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/).[2] The data set contains 1445 responses to the question "What aspects of your life have you considered when assessing your happiness?" Based on a coding manual, these responses were classified into 34 classes, such as family, mental health and job situation. We removed stopwords and stemming (in German). Then we converted the texts into unigram and bigram variables. Variables that appeared in two or fewer responses were removed as "rare terms". While it is a convenience sample, we make no claim that results are representative nor do we report substantative results. Our interest is merely in assessing the five strategies for automatic classification.

For automatic classification a statistical learning algorithm must be chosen. Here we fit SVMs with linear kernel because this choice is popular for text data (Joachims, 2001). We select the tuning parameter $C$ of SVMs through 10-fold cross-validation and allow the value of $C$ to be different for different strategies[3]. Then, we run 10-fold cross-validation for 100 times. The mean predictive accuracy of the 100 cross-validations is presented in Figure 3.

The Happiness data set has unordered classes while Patient Joe has ordered classes. After checking the coding matrices of the coders, the equal misclassification coding matrix $M_1$ appears reasonable.

To decide how many texts to code under a fixed budget, we need an estimate of the coding error rate. Since the coding error rate is unknown, we draw a random sample of 100 texts from each data set. We estimate the coding error rate $p$ to be 4% in the Happiness and 12% in the Patient Joe data. Based on these modest coding errors, we expect under a fixed budget single coding performs best and when double codes are already available "expert resolves" performs best.

We compare all strategies (except "majority vote") to verify our expectations. For the Happiness data (Figure 3), when double codes are available, "expert resolves" and "replicate" improve automatic coding significantly compared with single coding ($p = 0.025$ for "expert resolves" and $p = 0.030$ for "replicate"). Under a fixed budget, single coding performs significantly better than all the double coding strategies ($p < 0.001$ for each two-way comparison). While "replicate" performed better than expected, this is consistent with

[2]The Happiness data are available for replication purposes from Dr. Katharina Meitinger at k.m.meitinger@uu.nl.

[3]While tuning parameters were allowed to vary by strategy, in practice they were mostly constant. For the Happiness data the tuning parameter was always estimated as $C = 1000$. The only exception was for "replicate" strategy when data were already available, where it is $C = 500$. For the Patient Joe data, the tuning parameter was always estimated as $C = 100$.

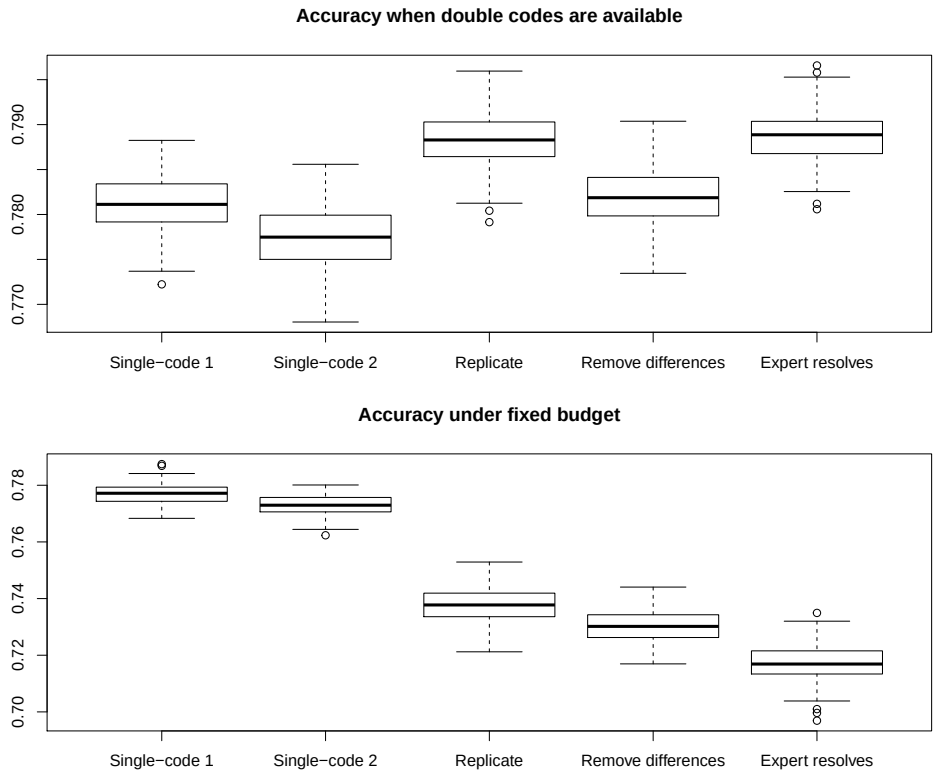


*Figure 3*. Boxplot of the predictive accuracy on the Happiness data when double codes are available (top) and under a fixed budget (bottom).

the results in Section 4.

For the Patient Joe data (Figure 4), we find that "expert resolves" and single coding works best when double coded data are available and under a fixed budget, respectively. When double codes are available, bootstrap tests show that the difference between single coding and "replicate" and between single coding and "expert resolves" are significant ($p = 0.011$ and $p < 0.001$, respectively). When the budget is fixed, "expert resolves" works significantly better than single coding ($p < 0.001$), "replicate" ($p < 0.001$) and "remove differences" ($p < 0.001$). This result is consistent with our expectation that single coding is preferable if the coding error rate is less than about 40%.

## 6 Discussion

Can double coding improve automatic coding of open-ended responses? We have evaluated four double coding strategies for multi-class classification. We found: 1) When double coded responses are available, use them. "Expert resolves" works best, followed usually by "majority vote" and, less often, "remove differences". 2) When the coding budget is fixed, use single coding unless error rates are very large. For large error rates (about 35% to 45% in most simulations), the double coding strategy "expert resolves" outperforms single coding. Remarkably, the findings are similar for different coding matrices.

For fixed cost in multi-class classification we found "majority vote" usually works better than "remove differences". For binary classification, He and Schonlau (2019) found the reverse. In multi-class classification knowledge that an observation likely belongs to one of two classes (the ones the coders disagree on) carries some information. In binary classification, a response with two different codes contains no information (There are only two classes; if coders choose one each, we do not learn anything). This explains why for multi-class classification "remove differences" is a less attractive strategy than for binary classification.

Throughout we have assumed that experts do not make errors. This assumption makes the simulations less complex. If we did allow for the expert to make errors, the performance of "expert resolves" would gradually get worse as a function of the assumed error. If the expert had an error as large as that of a regular coder, then the result would be the same as that of "majority vote" (assuming that the double coded data are available, i.e. we don't have to pay for the expert).

The limitations of the study include: 1) For statistical learning, we used SVM in our experiments. However, we also tried random forests (not shown) and obtained essentially the same results. We therefore do not believe results are

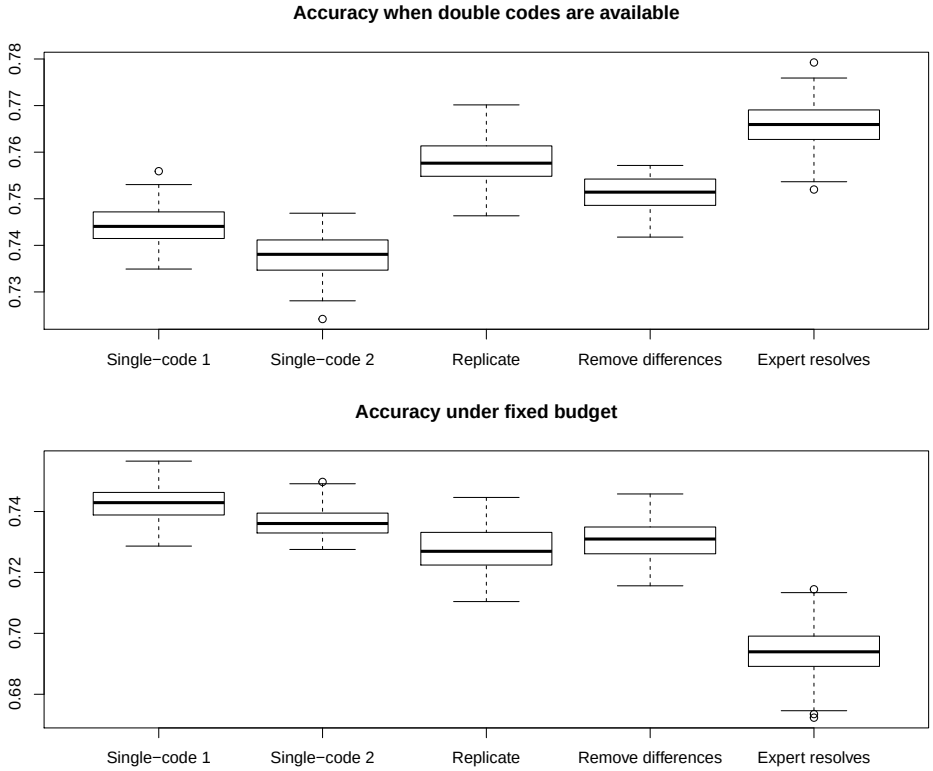


*Figure 4*. Boxplot of the predictive accuracy on the Patient Joe data when double codes are available (top) and under a fixed budget (bottom).

sensitive to the choice of the statistical learning algorithm. 2) We compared the performance of strategies based on predictive accuracy. Though accuracy is widely-used evaluation criterion, logloss is a smoother criterion for multi-class classification. In our experiments, the results based on accuracy and logloss are similar (results not shown). 3) For large error rates under a fixed budget the strategy changes. The threshold of what "large" constitutes is data dependent. In practice this does not matter much: if coding errors are large, you would want to redesign the coding procedure to reduce the coding error. For example, one might change or combine the answer categories or improving the coding manual. 4) We assume that regular coders have the same coding matrix. This is perhaps not true. However, assuming different coding matrices would further increase complexity and we have no reason to believe that it would make a difference in the conclusions.

We recommend the following for survey researchers who wish to code answers to open-ended questions automatically: when double coded texts are available and an expert has resolved differences, use the resolved coding. When double coded texts are available but no expert is available, use a third regular coder to resolve the differences. When double coded texts are not available and researchers have a fixed budget for manual coding, use single coding unless the error rate is very high (more than 35% – 45%). If the coding error rate is very high, one should probably redesign the coding strategy (redesign the manual, or the answer classes). Further, if the estimated probability of correct classification for some text answers is deemed too low, the concept of semi-automated classification (Schonlau & Couper, 2016) suggests to code those text answers manually.

## Acknowledgement

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

## References

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

Beatty, P. C., & Willis, G. B. (2007). Research synthesis: The practice of cognitive interviewing. *Public Opinion Quarterly*, *71*(2), 287–311.

Behr, D., Kaczmirek, L., Bandilla, W., & Braun, M. (2012). Asking probing questions in web surveys: Which factors have an impact on the quality of responses? *Social Science Computer Review*, *30*(4), 487–498.

Carley, K. (1993). Coding choices for textual analysis: A comparison of content analysis and map analysis. In P. Marsden (Ed.), *Sociological methodology* (Vol. 23, pp. 75–126). Oxford, Blackwell, UK.

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.

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*. January 1, 1997. doi:10.1787/18151981

Fleiss, J. L., Levin, B., & Paik, M. C. (2013). *Statistical methods for rates and proportions* (3rd). John Wiley & Sons, New York.

Gweon, H., Schonlau, M., Kaczmirek, L., Blohm, M., & Steiner, S. (2017). Three methods for occupation coding based on statistical learning. *Journal of Official Statistics*, *33*(1), 101–122.

He, Z., & Schonlau, M. (2019). Automatic coding of text answers to open-ended questions: Should you double code the training data? *Social Science Computer Review*. published online first at May 6, 2019. doi:10.1177/0894439319846622

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

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* (pp. 128–136). ACM, New Orleans, USA.

Kern, C., Klausch, T., & Kreuter, F. (2019). Tree-based machine learning methods for survey research. *Survey Research Methods*, *13*(1), 73–93.

Mannetje, A., & Kromhout, H. (2003). The use of occupation and industry classifications in general population studies. *International Journal of Epidemiology*, *32*(3), 419–428.

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*(sup3), 177–190.

Meitinger, K., Braun, M., & Behr, D. (2018). Sequence matters in web probing: The impact of the order of probes on response quality, motivation of respondents, and answer content. *Survey Research Methods*, *12*(2), 103–120.

Mullainathan, S., & Obermeyer, Z. (2017). Does machine learning automate moral hazard and error? *American Economic Review*, *107*(5), 476–80.

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

Schierholz, M. (2019). *New methods for job and occupation classification* (Doctoral dissertation, University of Mannheim). Retrieved from https://madoc.bib.uni-mannheim.de/50617/1/Dissertation_Schierholz.pdf

Schonlau, M. (2015). What do web survey panel respondents answer when asked "Do you have any other comment?" *Survey Methods: Insights from the Field*. 1-7. November 20, 2015. doi:10.13094/SMIF-2015-00013

Schonlau, M. (2020). Size text box—patient joe data. CentERdata. Retrieved from 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.

Schonlau, M., Guenther, N., & Sucholutsky, I. (2017). Text mining with n-gram variables. *The Stata Journal*, *17*(4), 866–881.

## Appendix A
## Derivations

### The Number of Texts under Fixed Budget for "Expert Resolves" and "Majority Vote"

Under a fixed budget of $N$ annotations, researchers can only afford to code limited texts. We first compute the expected cost of coding a single text under the strategy "expert resolves".

The probability that a text in class $i$ ($i$ is the true class) is coded differently by the two regular coders is $1 - \sum_{j=1}^{L} p_{ij}^2$, where $p_{ij}$ is the $(i, j)^{th}$ element in coding matrix $M$. So the probability that a random text is coded differently for the first two coders is $\sum_{i=1}^{L} q_i(1 - \sum_{j=1}^{L} p_{ij}^2)$, where $q_i$ is the marginal distribution of classes. Then, the average cost for coding a randomly picked text is $2 + t\sum_{i=1}^{L} q_i(1 - \sum_{j=1}^{L} p_{ij}^2)$, where $t$ denotes the relative cost of coding by an expert over a regular coder. Therefore, the number of texts under a fixed budget of N annotations is

$$\frac{N}{2 + t\sum_{i=1}^{L} q_i(1 - \sum_{j=1}^{L} p_{ij}^2)} = \frac{N}{2 + t - t\sum_{i=1}^{L} q_i \sum_{j=1}^{L} p_{ij}^2} \quad (1)$$

In terms of costs, "majority vote" can be viewed as "expert resolves" with $t = 1$. Therefore, the number of texts under a fixed budget follows from formula 1 by setting $t = 1$:

$$\frac{N}{3 - \sum_{i=1}^{L} q_i \sum_{j=1}^{L} p_{ij}^2} \quad (2)$$

## Appendix B
## Figures

**Experimental Results for the Coding Matrix with Misclassification in Higher Classes with Different Simulation Parameters**

In Section 4.3, we ran simulated experiments on the Patient Joe data using the coding matrix $M_3$ and showed experimental results when the parameters in $M_3$ was $g_1 = 0.2$ and $g_2 = 0.2$. In order to show that the result is not sensitive to the choice of $g_1$ and $g_2$, here we present results for the Patient Joe data when $g_1$ and $g_2$ take other values. Specifically, we consider three combinations: $g_1 = 0.2$ & $g_2 = 0.5$, $g_1 = 0.5$ & $g_2 = 0.2$, and $g_1 = 0.5$ & $g_2 = 0.5$.

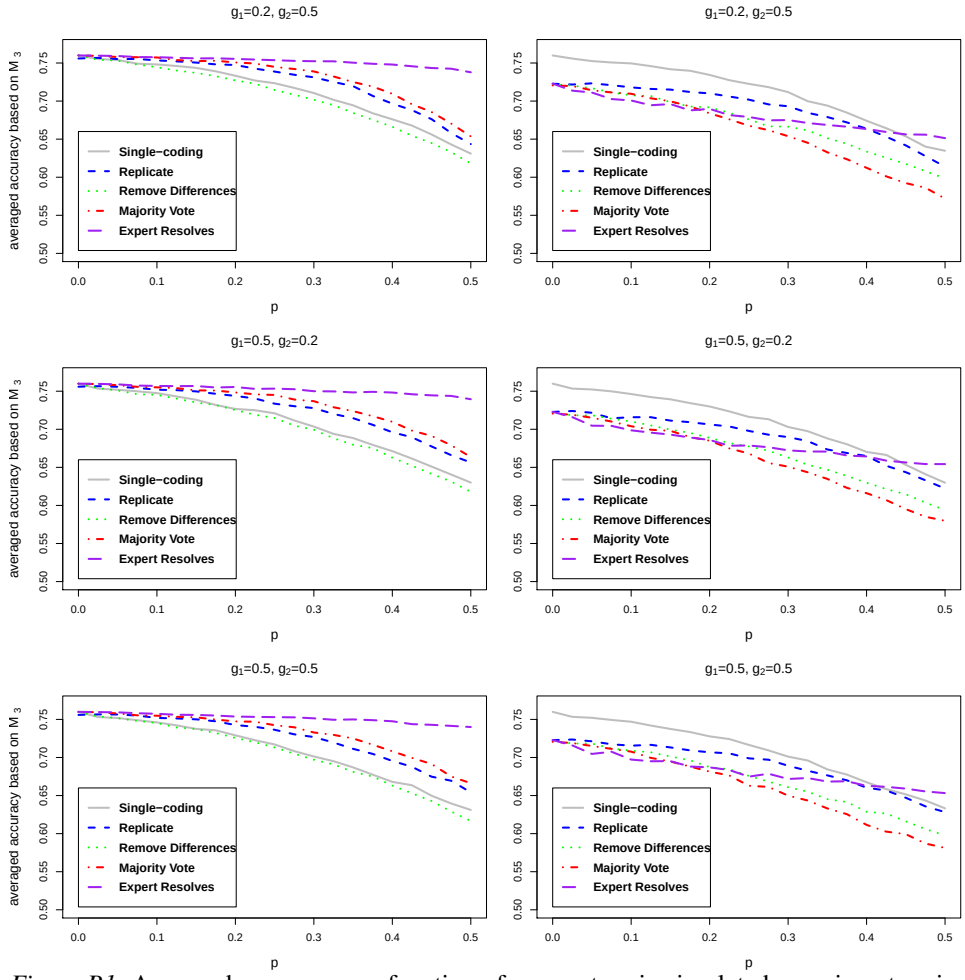


*Figure B1*. Averaged accuracy as a function of error rate $p$ in simulated experiments using the Patient Joe data, when we assume the coding matrix is like $M_3$. Top plots are for $g_1 = 0.2$ and $g_2 = 0.5$, middle plots are for $g_1 = 0.5$ and $g_2 = 0.2$, and bottom plots are for $g_1 = 0.5$ and $g_2 = 0.5$. The first column shows experiments when double coded data are available while the second column shows when the budget is fixed.

Figure B1 shows similar results as in Section 4.3. When double coded texts are available, "expert resolves" works better than single coding and other double coding strategies. Under a fixed budget, single coding is preferable unless the coding error rate is too high (> 45%). The different choices of $g_1$ and $g_2$ do not have a large influence on the results.