On the Automatic Coding of Text Answers to Open-ended Questions in Surveys

by

Zhoushanyue He

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Statistics

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Andy Peytchev, Senior Survey Methodologist RTI International
Supervisor(s):	Matthias Schonlau, Professor Dept. of Statistics and Actuarial Science, University of Waterloo
Internal Member:	Shoja'eddin Chenouri, Professor Dept. of Statistics and Actuarial Science, University of Waterloo
Internal Member:	Martin Lysy, Associate Professor Dept. of Statistics and Actuarial Science, University of Waterloo
Internal-External Member:	Sarah Wilkins-Laflamme, Assistant Professor Dept. of Sociology and Legal Studies, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Open-ended questions allow participants to answer survey questions without any constraint. Responses to open-ended questions, however, are more difficult to analyze quantitatively than close-ended questions. In this thesis, I focus on analyzing text responses to open-ended questions in surveys. The thesis includes three parts: double coding of open-ended questions, predictions of potential coding errors in manual coding, and comparison between manual coding and automatic coding.

Double coding refers to two coders coding the same observations independently. It is often used to assess coders' reliability. I investigate the usage of double coding to improve the performance of automatic coding. I find that, when the budget for manual coding is fixed, double coding which involves a more experienced expert coder results in a smaller but cleaner training set than single coding, and improves the prediction of statistical learning models when the coding error rate of coders exceeds a threshold. When data have already been double coded, double coding always outperforms single coding.

In many research projects, only a subset of data can be double coded due to limited funding. My idea is that researchers can make use of the double-coded subset to improve the coding quality of the remaining single-coded observations. Therefore, I propose a model-assisted coding process that predicts the risk of coding errors. High risk text answers are

then double-coded. The proposed coding process reduces coding error while keeping the ability to assess inter-coder reliability.

Manual coding and automatic coding are two main approaches to code responses to open-ended questions, yet the similarity or difference in terms of coding error has not been well studied. I compare the coding error of human coders and automated coders. I find, despite a different error rate, human coders and automated coders make similar mistakes.

Acknowledgements

I would like to thank my supervisor Dr. Matthias Schonlau. This thesis would not have been completed without his guidance.

I also thank members of my thesis comittee, Dr. Shoja'eddin Chenouri, Dr. Martin Lysy, Dr. Sarah Wilkins-Laflamme, and Dr. Andy Peytchev for serving as my committee members.

I deeply appreciate the help and support from my parents, Dr. Lin Yang and Mr. Zhi Dong in my doctoal study, especially during the hard time of COVID-19 pandemic.

I gratefully acknowledge LISS (Longitudinal Internet Studies for the Social sciences) panel administered by CentERdata, Canadian Cancer Society Smokers' helpline and Dr. Katharina Meitinger at University of Utrecht for allowing me using the data.

My PhD research has received financial support from Ontario Graduate Scholarship, Queen Elizabeth II Graduate Scholarship in Science and Technology, President's Graduate Scholarship, and teaching and research assistantships from the Department of Statistics and Actuarial Science. I am grateful for having this funding. I thank Dr. Mary Thompson and Mary Lou Dufton for their help in my application for the funding.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

An open-ended question refers to a question that cannot be answered using yes/no or options pre-specified by survey researchers. Instead, it allows survey participants to give whatever answers they want. Some open-ended questions are:

- who was involved in this project?
- What is your job?
- When did you have your first kid?

- Where did you first meet your partner?
- Why did you choose that answer to the previous question?
- How do you like the government's immigration policies?

There are many types of open-ended questions in surveys. Some questions ask for short text answers, while some others encourage respondents to give long answers. Unfortunately, there is no exhaustive list of different types of open-ended questions in the literature.

One type of open-ended questions in surveys is final comments. Final comments refer to the questions near the end of a survey asking whether participants have additional comments (Schonlau, 2015). McLauchlan and Schonlau (2016) analyzed final comments in a longitudinal study and found shorter comments are associated with increased next-wave attrition while longer comments are associated with decreased next-wave attrition.

Another type of open-ended questions is probing questions. Probing questions are follow-up questions asking respondents to provide additional information about a survey item (Beatty and Willis, 2007; Meitinger et al., 2018). Behr et al. (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.

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 own 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 a closed-ended question failed to capture many dimensions.

There are many factors affecting responses to open-ended questions. For example, Engwall (1983) found respondents with different demographic composition have a different proportion of positive and negative responses to a question. Also, the design of questions influences how participants answer these questions. Gendall et al. (1996) investigated the effect of a question itself on the length and content of the responses; They found that a negative cue produced the most negative responses, a positive cue produced the most positive responses, and a neutral cue produced the most neutral responses. Brennan and Holdershaw (1999) extended the research and found that longer responses and a greater number of ideas can be elicited by using different cue tones in separate questions rather than combining them in a single long question. They also found, for combined questions, the ordering of the cue tones had a pronounced effect on the tone of the ideas elicited, but not on the total number of ideas generated.

Responses to open-ended questions in surveys are often text data. Thus, open-ended

responses are usually more difficult for quantitative analysis than numeric data because they are unstructured. A common way to analyze text answers is to classify them into categories. For example, occupation coding is to code answers of an open-ended question about one's job. The classification of text answers can be either manual or automatic (or both).

1.1.1 Manual Coding

Manual coding involves one or more human coders (Roberts et al., 2014), and these coders assign appropriate codes based on some coding guidance or a codebook developed by a designer (Esuli and Sebastiani, 2010). When more than one coders code the same texts, it is natural that different coders have different opinions on some texts (Conrad et al., 2016), which may be due to ambiguity of texts, lack of clarity of the coding manual, or different personal understandings. Inter-coder disagreements are common in practice (Crittenden and Hill, 1971; Ames et al., 2005). Popping and Roberts (2009) discussed typical sources of disagreement among coders such as differences in coders' identifications of clauses and disagreements in identification. They also pointed out the need to resolve discrepancies among coders.

Inter-coder disagreement/agreement is often used as a diagnostic tool for the reliability of the coding procedure. Intercoder reliability is a measure of agreement between multiple

coders about how they apply codes to the data (Kurasaki, 2000). Researchers typically measure intercoder reliability using Cohen's Kappa coefficient, a measure that takes into account chance agreement among the human coders (Fleiss et al., 2013). In large data sets, often only a subset of the data is double coded to determine Kappa; the rest is single coded. Inter-coder reliability refers to the double-coded texts; it does not change anything for the single-coded texts or inform the coding of uncoded texts.

Many studies have suggested that the level of inter-coder reliability is low for some coding tasks (Montgomery and Crittenden, 1977; Schonlau, 2015). 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 disagreements can be resolved in one of several ways, including: 1) The two coders discuss the disagreements and reach a consensus. 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.

1.1.2 Automatic Coding Using Statistical Learning Models

Automatic coding refers to using statistical learning methods to code text answers. Popular statistical learning methods applied in analyzing open-ended questions include Naïve Bayes (Severin et al., 2017), support vector machine (SVM) (Bullington et al., 2007) and

tree-based methods (random forests, boosting) (Kern et al., 2019). For example, Joachims (2001) developed a text classification model based on support vector machines and achieved better classification performance than conventional generative models. Gweon et al. (2017) proposed three automatic coding methods for occupation coding and showed they improved coding accuracy. Some researchers combined statistical learning algorithms with manual coding to achieve better classification. Schonlau and Couper (2016) proposed a semi-automatic algorithm based on multinomial gradient boosting to code text answers automatically if automatic coding was likely code correctly and manually otherwise.

In this thesis, I use two widely used statistical learning models, support vector machines (SVMs) and random forests (RF). 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 points 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.

A general process to assess the performance of statistical learning models on a set of texts is cross-validation. One of the cross-validation methods, Holdout method, randomly

divides the whole data set into a training set and a test set. The models of interest are applied to the training set, and the predictions of labels on the test set are made based on the fitted model. Then researchers can evaluate how the fitted model works by comparing the predictions with the true labels of the test texts (Lewis and Ringuette, 1994; Bijalwan et al., 2014).

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 other data (test data) (Lewis and Ringuette, 1994; Bijalwan et al., 2014). 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 the 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.

1.1.3 Manual Coding vs. Automatic Coding

The comparison between manual coding and automatic coding is in two dimensions: coding reliability and coding cost.

Weber (1990) proposed three components in coding reliability:

- Stability: the ability of a coder to consistently assign the same code to a given text.
- Reproducibility: intercoder reliability.
- Accuracy: the ability of a group of coders to conform to a standard.

Automatic coding provides at least one advantage over human coding in terms of coding reliability: stability. A trained model does not change its classification, yet a human coder may change his/her opinion towards a given text in the coding process, either consciously or unconsciously. Moreover, manual coding process is considerably subjective - different coders may have different opinions on a given text - whereas automatic coding is not prone to inconsistencies (Patel et al., 2012). The reproducibility of automatic coding and human coding is somehow hard to compare because intercoder reliability usually changes case by

case. Human coders usually have higher accuracy than automatic coding, partly because human coders can read, understand, and classify even particularly difficult answers.

The cost of automatic coding and human coding depends on the size of the data set. For a small set of text responses, automatic coding requires a (manually) coded training set of proper size (which may not be plausible in this case). Therefore, applying manual coding to a small data set seems more cost-efficient. Things are different for a large data set. Manual coding tends to be expensive and time-consuming as human coders have to read observations one by one (Geer, 1991; Grimmer and Stewart, 2013). On the contrary, once a statistical learning model is trained, it costs almost nothing to code an additional observation.

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 misunderstanding of the meaning of responses (Funkhouser and Parker, 1968; He and Schonlau, 2020a). Researchers have emphasized the need to assess and improve coder reliability (Crittenden and Hill, 1971; Kassarjian, 1977; Montgomery and Crittenden, 1977; Hughes and Garrett, 1990). Lombard et al. (2002) provided a standard guideline regarding the procedure for assessing and reporting inter-coder reliability. The coding error of automated coders comes from different sources such as errors in the training data (Belloni et al., 2016) and generalization (out of sample) error of the fitted model (Giorgetti et al.,

2003).

Despite the widespread application of statistical learning, there are relatively few studies about the similarity and difference between classifying text answers using statistical learning models and classifying by humans, nor how manual coding and automatic coding can improve the performance of each other. Conway (2006) pointed out that using computer-assisted coding allowed researchers to avoid problems with inter-coder reliability, a major issue of human coding when multiple coders are involved. However, whether humans and models make similar coding errors has not yet been addressed in the literature.

1.2 Summary of Contributions

The first two contributions (He and Schonlau, 2020a,b) investigate whether double coding the training set of text responses can help improve the performance of automatic coding. I develop strategies for using double-coded data for automatic coding. I compare these strategies with single coding in two scenarios: 1) when the double codes are available so that there is no need to consider coding cost, and 2) when the training data have not been coded and the budget for coding is fixed. Simulations show that double coding outperforms single coding when the cost is not a concern. When the budget is fixed, double coding helps improving the performance of automatic coding if human coders have a high coding

error rate. In addition to the content covered by the two papers, Section 2.6 discusses a technique to compare single coding and double coding in a small-sized experiment.

When coding a large set of text answers, a routine procedure is to double code a subset of the data to assess coding reliability with other observations single coded. Although disagreements in the double-coded subset may indicate coding mistakes, researchers are unaware of coding errors in the single-coded observations. My third contribution (He and Schonlau, ND) is to reduce coding mistakes in the single-coded observations using the double-coded subset. I propose a model-assisted coding procedure, in which a fitted model is used to predict the risk of disagreement. Observations with high predicted risk are double coded. The advantage of the proposed coding procedure is that it double codes "hard-to-code" observations while keeping the ability to assess inter-coder reliability.

The forth contribution (He and Schonlau, to appear) explores whether and to what extent coding errors from manual coding and automatic coding differ. Both manual coding and automatic coding make mistakes but the sources of mistakes are different. I compare the mistakes made by human coders and the mistakes made by models and find that human coders and models tend to find the same text answers difficult to code. Also, there appears to be no point to have more than one model to investigate coding differences, while having multiple human coders is beneficial.

1.3 Summary of Data Sets

Four sets of text responses to open-ended questions are used in the thesis: Patient Joe, Smokers' Helpline, Happiness and Democracy. The Patient Joe, Happiness and Democracy data sets were initially double coded by two coders, and disagreements between the two coders were resolved by either an experienced expert (Patient Joe data) or a group discussion with other researchers (Happiness and Democracy data). The resulting coding is called "gold standard coding" (the best classification we can get in practice). The Smokers' Helpline data set was single coded. Chapter 2 uses the Patient Joe and Smokers' Helpline data sets. Chapter 3 and 4 use the Patient Joe, Happiness and Democracy data sets.

The Patient Joe data set (Schonlau, 2020) contains 1,758 answers to the following long-answer 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 (`http://www.lissdata.nl`) in 2012. The answers in this data set have been classified into one of four ordered classes: proactive, somewhat proactive, passive and destructive (Schonlau and Couper, 2016). In Section 2.3.1, the data set is converted into a

binary classification problem of whether a response is proactive or not. The coding manual of the Patient Joe data is in Appendix A.

The Smokers' Helpline dataset came from the 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 was a follow-up phone survey during which the following short-answer open-ended question was asked "What helped you the most in trying to quit (smoking)?". The Smokers' Helpline data set contains a total of 3,352 observations. Responses were recorded and manually coded into one of 27 categories. In Section 2.3.1, I consider a binary classification on whether the willpower helped respondents the most in trying to quit smoking.

Both the Happiness and Democracy data sets were collected in German as part of a web survey conducted in Germany in 2017. The participants were chosen from respondi's German online-access panel (`http://www.respondi.com/EN/`). The Happiness data set contains 1,445 answers to the short-answer question "What aspects of your life did you considered when assessing your feeling of happiness?" ("An welche Aspekte lhres Lebens haben Sie bei der Beurteilung lhres Glücksgefühls gedacht?"). The Democracy data set contains 1,096 answers to the probing question "What aspects did you think of when answering this question?" ("An welche Aspekte haben Sie bei der Beantwortung der Frage gedacht?"), which referred to the earlier question about democracy in Germany

("Wie zufrieden sind Sie - alles in allem - mit der Art und Weise, wie die Demokratie in Deutschland funktioniert?"). The researchers under the leadership of Professor Meitinger at the University of Utrecht classified the Happiness and Democracy data sets. The coding schemas for the two data sets are documented in Appendix B and C, respectively. They contain the original questions, coding instructions and different levels of classes. In the thesis, I use the aggregated categories in the coding schemas. The Happiness data set contains 10 aggregated classes: "social network & surrounding", "health", "job", "financial situation", "life situation & living conditions", "politics, security & society", "life event", "time references", "rest" and "problems & nonresponse". The Democracy data set has 7 agggregated classes: "akteur & gruppen", "politikfelder", "situation", "beurteilung verhalten politiker & parteien", "demokratische system", "rest" and "problems & nonresponse". An aggregated class may include multiple categories of lower levels. For example, in the Happiness data set, the aggregated class "health" contains "general health" (coded as "2" in the coding schema), "physical health" (coded as "21") and "mental health" (coded as "22"). Similar goes for the Democracy data: For instance, the aggregated class "akteur & gruppen" contains codes 101, 102, 103, 104 and 105.

For all the data sets, I constructed so-called n-gram variables (Büttcher et al., 2016). A 1-gram (or unigram) variable is a variable that counts how often a given single word occurs in an answer text. A 2-gram (or bigram) variable is a variable that counts how often a given

Data	Data Size	Size of Training Set	Size of Test Set	Number (Percent) of Disagreements	Used in Chapter
Patient Joe	1756	1000	756	407(23.2)	2, 3 and 4
Smokers' Helpline	3352	2000	1352	N/A	2
Happiness	1438	800	638	83(5.8)	2, 3 and 4
Democracy	1096	600	496	158(14.4)	3 and 4

Table 1.1: Some Details of the Patient Joe, Smokers' Helpline, Happiness and Democracy data sets.

sequence of two words occurs in an observation. Also, I removed stopwords (commonly occurring stopwords such as "the" and "a") and used stemming (truncating words so that there is only one variable "walk" for variations like "walking", "walks" and "walked") in English for the Patient Joe and Smokers' Helpline and Dutch for the Happiness and Democracy data sets. Table 1.1 summarizes the sizes of the four datasets as well as the numbers and percentages of inter-coder disagreements. Note that the Smokers' Helpline data were not double coded, thus there is no information about inter-coder disagreements for the Smokers' Helpline data set. Also, the sizes of the training set and the test set in Table 1.1 are applied to Chapter 2 and 4. Chapter 3 adopts a different data split.

Chapter 2

Automatic Coding of Text Answers: Whether and How to Use Double Coded Data

2.1 Introduction

To analyze text data collected from open-ended questions quantitatively, researchers often classify these data into pre-specified classes. Traditionally, text data are 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., 2019; Ye et al., 2018; Matthews et al., 2018). Statistical or machine learning algorithms are generally gaining in popularity in text classification (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.

Because automatic coding predicts the classes of texts based on a trained model, the quality of automatic coding depends on the model, and the model relies heavily on manually coded data on which it is trained. Unfortunately, human coders may make mistakes due to human error but also because ambiguous texts are difficult to code. A learning model of training data with coding error is likely to perform worse than one without coding error.

To learn about the degree of manual coding disagreement, a common practice is to double code: each text is coded by two human coders, and each coder codes without reference to the other (Elias, 1997). Double coding allows assessing how much the two human coders agree. If the two coders disagree on a large proportion of observations, researchers may need to modify the coding book and coding process to improve the coding reliability.

However, it is unknown whether and how a statistical learning model could benefit from double-coded data. In this study, four double coding strategies are proposed, and I

compare double coding and single coding with respect to their ability to improve automatic coding in both binary and multi-class settings.

The outline is as follows: Section 2.2 proposes strategies for resolving inter-coder disagreement in double coding. Section 2.3 investigates which strategy leads to the highest classification accuracy on simulated data in two scenarios: one is that we have a fixed coding budget and texts are not yet coded, and the other is that texts have already been double coded. Section 2.4 explores the sensitivity of the results with respect to the cost of an expert coder and simulation parameters. Section 2.5 compares the performance of these strategies based on double-coded data. Section 2.6 introduces a resampling technique helping researchers decide which strategy to use by a small-sized experiment. Section 2.7 discusses the implications and limitations of the study.

2.2 Strategies for Resolving Inter-Coder Disagreement in Double Coding

The coding performance of a coder can be represented using a coding matrix. 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 $\sum_{j=1}^{L} p_{ij} = 1$ for $\forall i \in \{1, 2, ..., L\}$.

In the following sections, I first consider a special case $p_{11} = 1 - p_{21}$ for $L = 2$ (binary classification), followed by general cases for arbitrary L (multi-class classification). The coding matrix for the special binary classification case can be written as

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

where p is the error rate of regular coders.

When two coders classify a response independently, they may assign different codes. In order to train statistical learning models, we need to resolve conflicts in the coded data. I consider the following strategies to resolve any inter-coder disagreement:

- Single coding: keep one coder's codes and discard the other coder's.
- Replicate: replicate each double-coded text into two observations, one with each of

the double codes, no matter whether the double codes are the same or not.

- Remove differences: remove text observations which are coded differently by the two coders from the data.

- Majority vote: if a text is coded differently by the two coders, a third coder codes. For simplicity, I assume the third coder can only choose a code from the first two codes, and the probability he/she selects a code is proportional to the corresponding probability in the coding matrix. Thus, the third code leads to a 2:1 majority.

- Expert resolves: an expert coder arbitrates any inter-coder disagreement. It is assumed in the study that the expert is always correct (although not literally true, this assumption approximates experts having higher coding accuracy).

Here the single coding strategy is equivalent to the common single coding of classifying text answers using one human coder.

The number of observations in the training data is different after applying different strategies. 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 may lead to different costs for coding an observation. 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' codes) without considering the coding cost. 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 because each observation requires two annotations (an annotation is the workload that a regular coder codes an observation). The cost of "majority vote" or "expert resolves" is more than twice that of single coding in that a third coder or an expert coder needs to be employed if the two coders disagree. "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 double coding strategies since duplicating or removing an observation requires no additional coding.

Therefore, the number of text 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, for "replicate" and "remove differences", the number under a fixed budget is

$$N/2 \tag{2.1}$$

To calculate the number of texts under a fixed budget for "majority vote" and "expert

resolves", the expected cost of coding a single text under the strategy "expert resolves" has been computed: Supposing the true class of a text is i, the probability it is coded differently by the two regular coders is $1 - \sum_{j=1}^{L} p_{ij}^2$. whether p_{ij} is the $(i,j)^{th}$ element in the coding matrix M. Then the probability that random text is coded differently by the two coders is

$$\sum_{l=1}^{L} q_l(1 - \sum_{j=1}^{L} p_{lj}^2) = 1 - \sum_{l=1}^{L}\sum_{j=1}^{L} q_l p_{lj}^2 \tag{2.2}$$

Let t denote the relative cost of coding by an expert over a regular coder. The cost for coding a text using "expert resolves" is two annotations (by the two regular coders) plus an additional cost if the two coders disagree. In other words, the cost for using "expert resolves" to code a text is

$$E(cost_{ER}) = 2 + t - t\sum_{l=1}^{L}\sum_{j=1}^{L} q_l p_{lj}^2 \tag{2.3}$$

Similarly, the average cost for using "majority vote" is

$$E(cost_{MV}) = 3 - \sum_{l=1}^{L}\sum_{j=1}^{L} q_l p_{lj}^2 \tag{2.4}$$

Therefore, when we have a fixed budget of N annotations, the number of texts we can

afford (on average) is

$$\frac{N}{2 + t - t\sum_{l=1}^{L}\sum_{j=1}^{L} q_l p_{lj}^2} \tag{2.5}$$

for "expert resolves" and

$$\frac{N}{3 - \sum_{l=1}^{L}\sum_{j=1}^{L} q_l p_{lj}^2} \tag{2.6}$$

for "majority vote".

The general coding matrix M contains $L(L-1)$ parameters. In practice, the coding matrix is unknown and contains too many parameters to estimate. In this thesis, I 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 classification represents the case where coding error happens at random with equal probabilities. It could serve as a proper default choice. The other two coding matrices I 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 into a higher class. The above coding matrices are chosen for their simplicity and practicability. More complex coding matrices exist, of course. For a specific data set, researchers may decide which special case fits the problem at hand.

2.2.1 Multi-class Coding Matrix 1: Equal Misclassification Probabilities

In the first special case, 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 a coding error rate p, and he/she is equally likely to classify a response into any incorrect class if a mistake happens.

Assuming the coding matrix is M_1 and using formulas 2.1, 2.5 and 2.6, the number of texts that can be coded under a fixed budget of N annotations is listed in Table 2.1, which also contains cases for some specific values of the error rate p. Unlike the general formulas 2.5 and 2.6, the formulas in Table 2.1 do not depend on the marginal class distribution $\{q_i\}_{i=1}^{L}$.

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	$\frac{N}{2+2tp-tp^2L/(L-1)}$	$\frac{N}{2+0.2t-0.01tL/(L-1)}$	$\frac{N}{2+0.4t-0.04tL/(L-1)}$

Table 2.1: Number of texts coded under a fixed budget of N annotations when the coding matrix is M_1.

2.2.2 Coding Matrix 2: Misclassification in Neighboring Classes

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

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

The matrix suggests that a coder has a probability of p to incorrectly classify an observa-

tion into a neighboring class, and if there are two neighboring classes, the probability of classifying into any of them is equal (i.e. $p/2$).

2.2.3 Coding Matrix 3: Misclassification in Higher Classes

The third special case I 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 observations into "higher" classes. The parameters $g_1, g_2, \dots, g_{L-2}$ show the strength of the tendency. An example of the personal tendency is that an optimistic coder may consider responses to be in more "optimistic" classes.

2.3 Comparing Strategies in Simulations

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

For automatic classification, a statistical learning algorithm must be chosen. Here I fit support vector machines (SVMs) with a linear kernel because this is a popular choice for text data (Joachims, 2001). I use the accuracy of automatic coding as the evaluation criterion for comparing the strategies. Accuracy is defined as the proportion of correctly coded observations, i.e. the text responses of which the predicted classes match their true classes.

2.3.1 Binary Classification with Equal Error Rate

Before looking at the three special coding matrices for general L, the binary classification when $L = 2$ is a good and simple start. I have shown in Section 2.2 that if there are only two classes and coders have equal probability p to code a text in one class incorrectly to the other, the coding matrix is

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

For both the Patient Joe and Smokers' Helpline data sets, unigram and bigram variables that did not appear in at least 5 texts are removed. Each data set is randomly split into a training set and a test set, with the sizes specified in Table 1.1. I simulate regular coders' codes from the gold standard codes by randomly changing the correct codes to the incorrect codes with probability p, where p is the error rate of the simulated coders. That is, for each observation I 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.

As I have pointed out previously, different strategies for double coding result in different costs 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 needs 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 coding budget is fixed, the number of 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.

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

I assume our expected coding budget allows single coding the whole training set, i.e., 1000 annotations for the Patient Joe data and 2000 annotations for the Smokers' Helpline data. We either single code the whole training set or double code a random subset of the training set (the subset size can be calculated using the foregoing formulas).

Figure 2.1 shows the average prediction accuracy for the five strategies – single coding and the four double coding strategies – as a function of the coding error rate for both datasets when the budget is fixed. The prediction accuracy is averaged over 100 repeated simulations. Because there are two classes, an error rate of 50% corresponds to random guessing and an error rate over 50% means human coding is worse than random guessing. The plot suggests that when the expected coding budget is fixed: 1) As the coding error

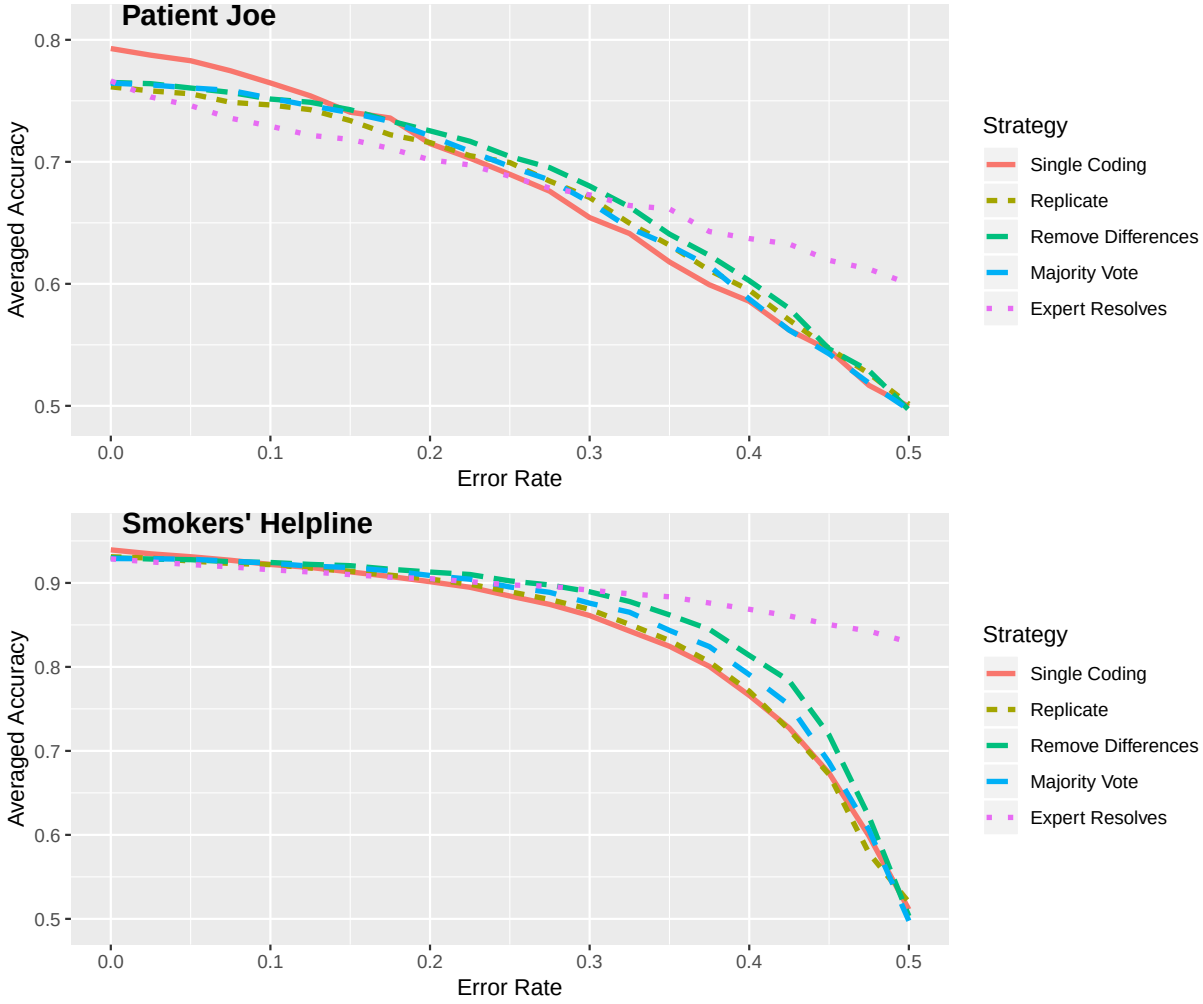


Figure 2.1: Average accuracy as a function of error rate p in simulations using the Patient Joe and Smokers' Helpline data sets when the expected coding budget is fixed.

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 2.2 shows the average accuracy of predictions from fitted models in 100 repeated simulations on the Patient Joe and the Smokers' Helpline, when the training data have already been double coded. 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 rate gets large but remains below random guessing (50% error rate); 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 even slightly better than "majority vote". 5) "Replicate" performs worst among the four double coding strategies.

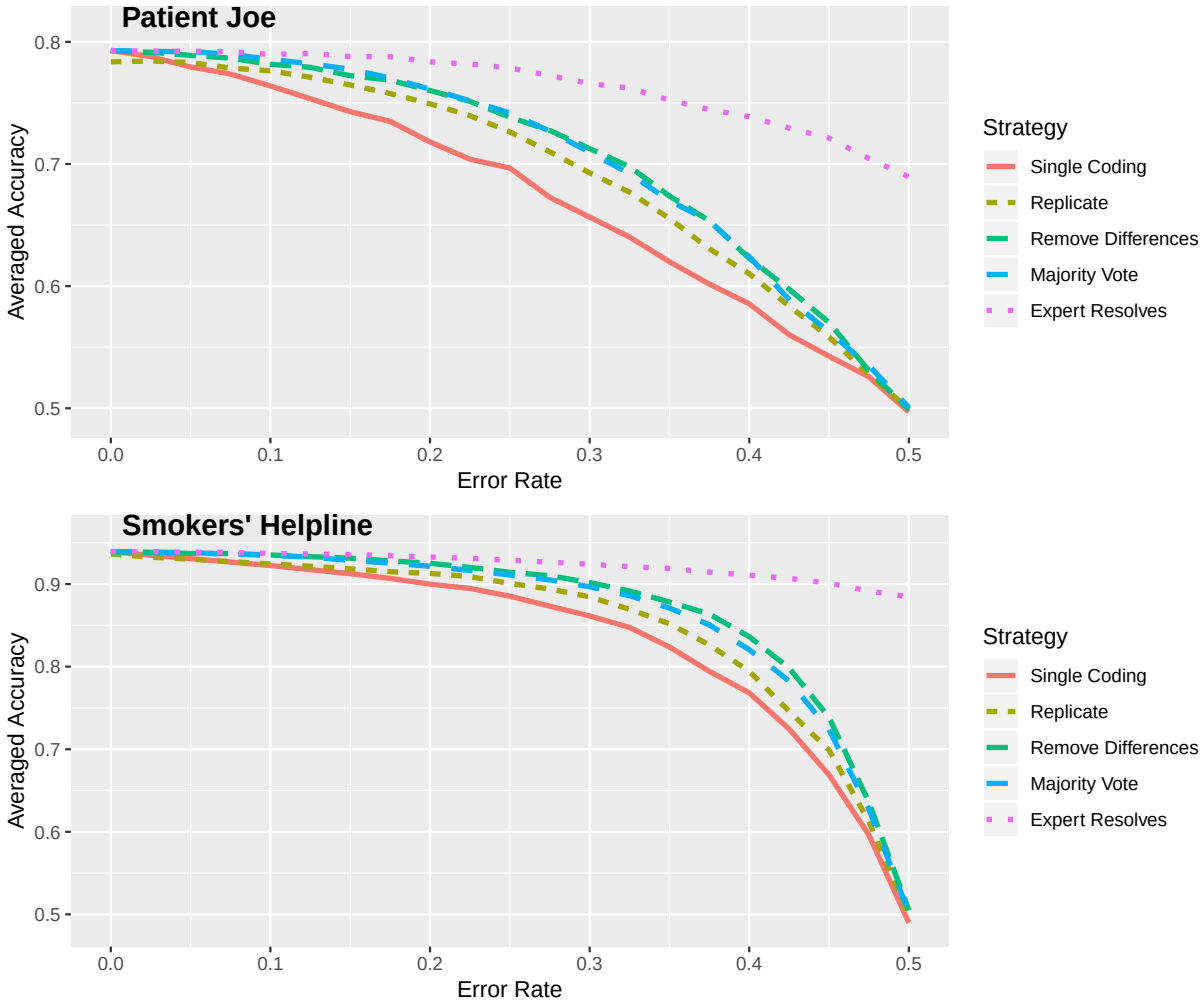


Figure 2.2: Average accuracy as a function of error rate p in simulations using the Patient Joe and Smokers' Helpline data sets when the data have already been double coded.

2.3.2 Special Coding Matrices for Multi-class Classification

To explore which coding strategy works best in multi-class classification, I use the three special coding matrices proposed in Section 2.2 and run simulations based on the Patient Joe data set. Under a fixed budget, the size of training set is calculated using formulas 2.1, 2.5 and 2.6 with $N = 1000$.

Figures 2.3a and 2.3b show the average predictive accuracy as a function of the error rate p for various strategies, when the coding matrix of a regular coder is M_1. For each value of p, the simulation is repeated 100 times.

When the double coded texts are already available (Figure 2.3a), "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 2.3b), no single strategy dominates: for low error rates single coding is best, and for high error rates "expert resolves" is best. The threshold for the transition is about 35%.

Assuming the coding matrix is M_2, Figures 2.3c and 2.3d show the predictive accuracy as a function of the error rate p averaged over 100 repeated simulations. Unlike for coding matrix M_1, a marginal distribution of the classes need to be assumed for the simulation (There was no need to do so for M_1 because the results in Table 2.1 did not depend on the

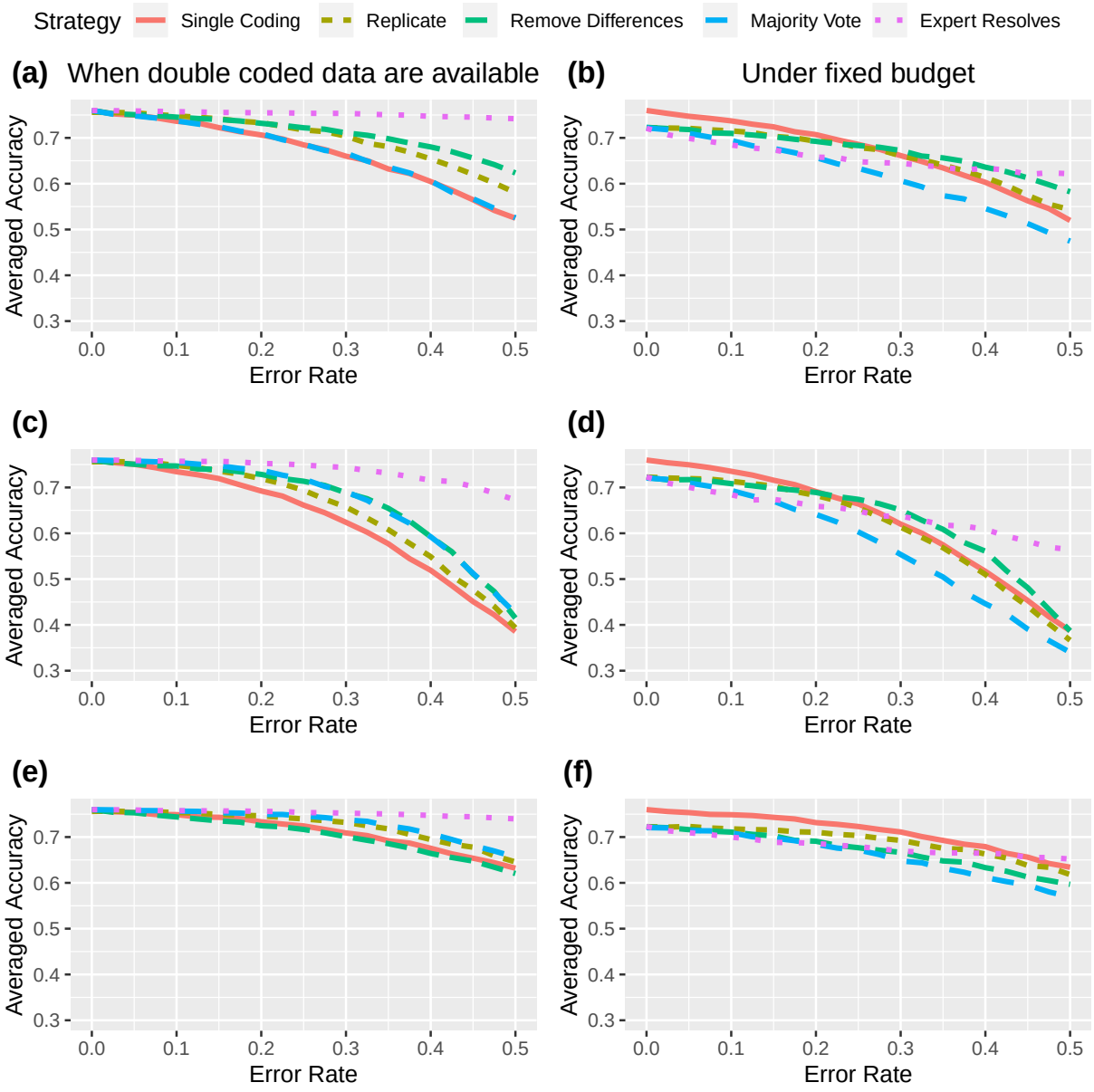


Figure 2.3: Average accuracy as a function of error rate p in simulations using the Patient Joe data. Each row represents a different coding matrix (M_1, M_2 and M_3). The coding matrix M_3 has parameters $g_1 = 0.2$ and $g_2 = 0.2$. The first column shows the results when double coded data are available, while the second column shows the results when the budget is fixed.

marginal distribution q_i). I assume the marginal distribution of classes is distribution 1 in Table 2.2.

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

Table 2.2: Assumed class distributions for the Patient Joe data

In Figure 2.3c, I observe a similar pattern as I have seen for coding matrix M_1. "Expert resolves" is the best strategy when double-coded texts are already available. In Figure 2.3d, 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.

Constrained to the misclassification in higher classes coding matrix M_3, 100 repeated simulations on the Patient Joe data with simulated coding are also run. The average predictive accuracy as a function of the error rate p is shown in Figures 2.3e and 2.3f. I also assume the marginal distribution of classes is distribution 1 in Table 2.2.

The parameters $\{g_i\}_{i=1}^{L-2}$ are simulation parameters that represent the tendency of a coder to consider a response in a "higher" class. In the Patient Joe data, $L = 4$. I 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 simulation results with other combinations of g_1 and g_2 are in Section 2.4.3. The results are similar.

For coding matrix M_3, I 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 simulations, 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".

2.4 Robustness Analysis

The simulations in previous sections involve some parameters: the relative cost of an expert coder over a regular coder t, the marginal class distribution $\{q_i\}$ for $i = 1, 2, ..., L$, and g_1 and g_2 in coding matrix M_3. To validate the results are robust, I re-run the simulations with different values of parameters.

2.4.1 Robustness of the Cost of Coding by an Expert

If the cost of an expert is lower than what I assumed in Section 2.3, hiring an expert coder becomes more cost-efficient and the "expert resolves" strategy becomes more preferable for lower error rates. I analyze how the relative cost of coding by an expert changes the threshold error rate at which "expert resolves" starts to predict most accurately in binary classification. For multi-class cases, the result is similar.

I investigate the relationship by performing simulations as a function of the relative cost t. Figure 2.4 shows, under a fixed budget, how the threshold error rate changes as the relative cost of coding by an expert 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 (> 15), "expert resolves" still beats single coding when coders make many mistakes (e.g. error rate $> 25 \sim 35\%$).

2.4.2 Robustness of the Marginal Class Distribution

Because coding matrices M_2 and M_3 depend on the marginal class distributions, using incorrect class distribution may lead to inaccurate estimation of the number of texts that can be coded under a fixed budget. I investigate the sensitivity of the results using different

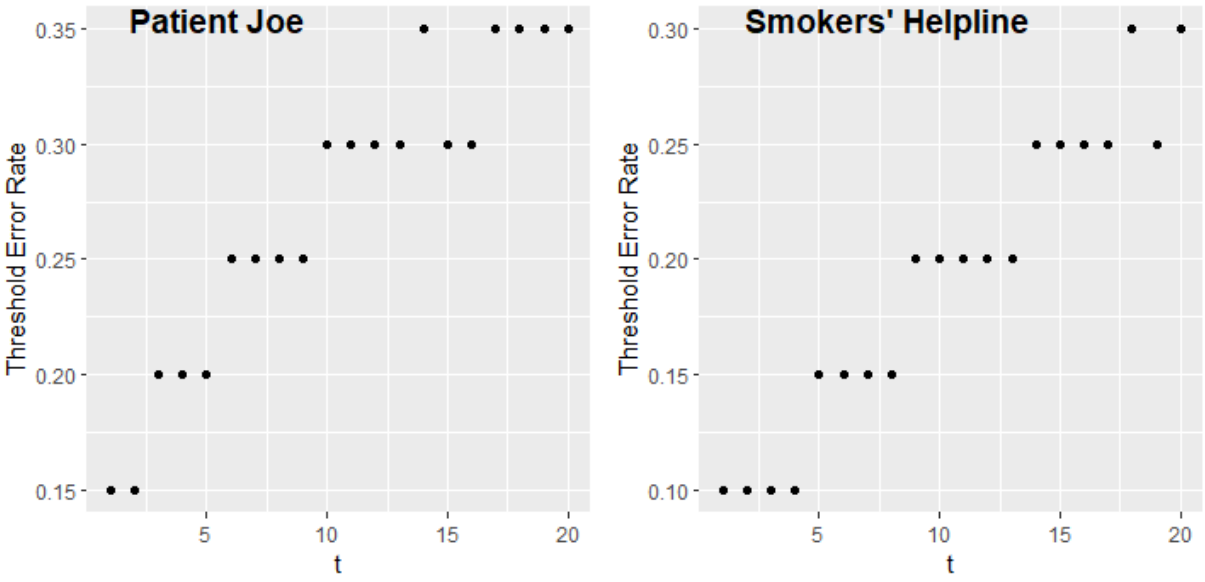


Figure 2.4: Threshold error rates that "expert resolves" outperforms single coding vs. the relative cost of coding by an expert t in binary classification. For a specific t, if the coding error rate is less than the threshold error rate, single coding results in more accurate predictions than "expert resolves"; if the error rate is larger than or equal to the threshold error rate, "expert resolves" predicts better than single coding.

class distributions. Specifically, we assume the classes are almost uniformly distributed (distribution 2 in Table 2.2).

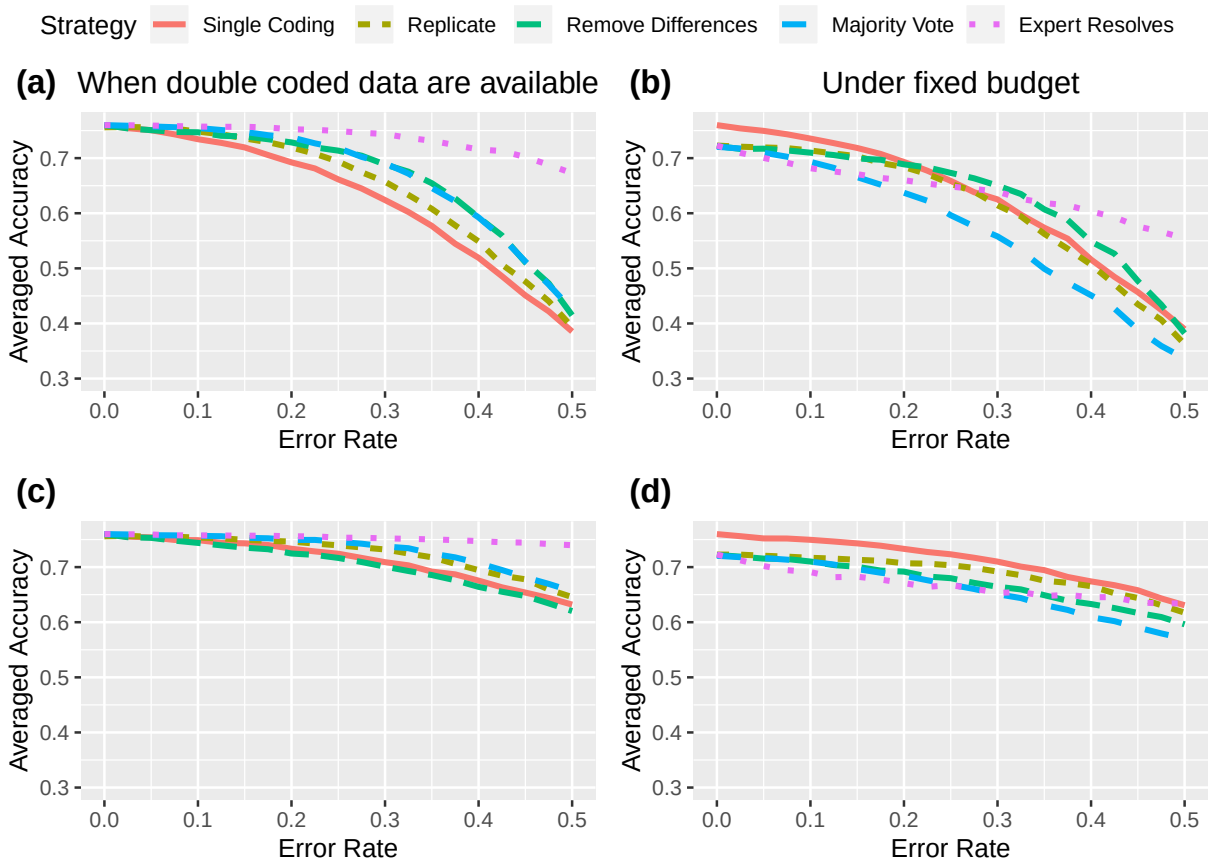


Figure 2.5: Sensitivity analysis for the Patient Joe data with different marginal class distributions. Otherwise it is analogous to Figure 2.3.

Figure 2.5 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 curves is the same, using a more uniform distribution of classes increases the threshold be-

tween single coding and "expert resolves". This probably does not have much impact in practice: if the coding error is large, the coding procedure should be redesigned.

2.4.3 Results for the Coding Matrix with Misclassification in Higher Classes with Different Parameters

In Section 2.3.2, I run simulations on the Patient Joe data using the coding matrix M_3 and show the simulation results when the parameters in M_3 is set to be $g_1 = 0.2$ 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 I present the results for the Patient Joe data with different g_1 and g_2. Specifically, I 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 2.6 shows the result of the three combinations, which are similar to those in Section 2.3.2. 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.

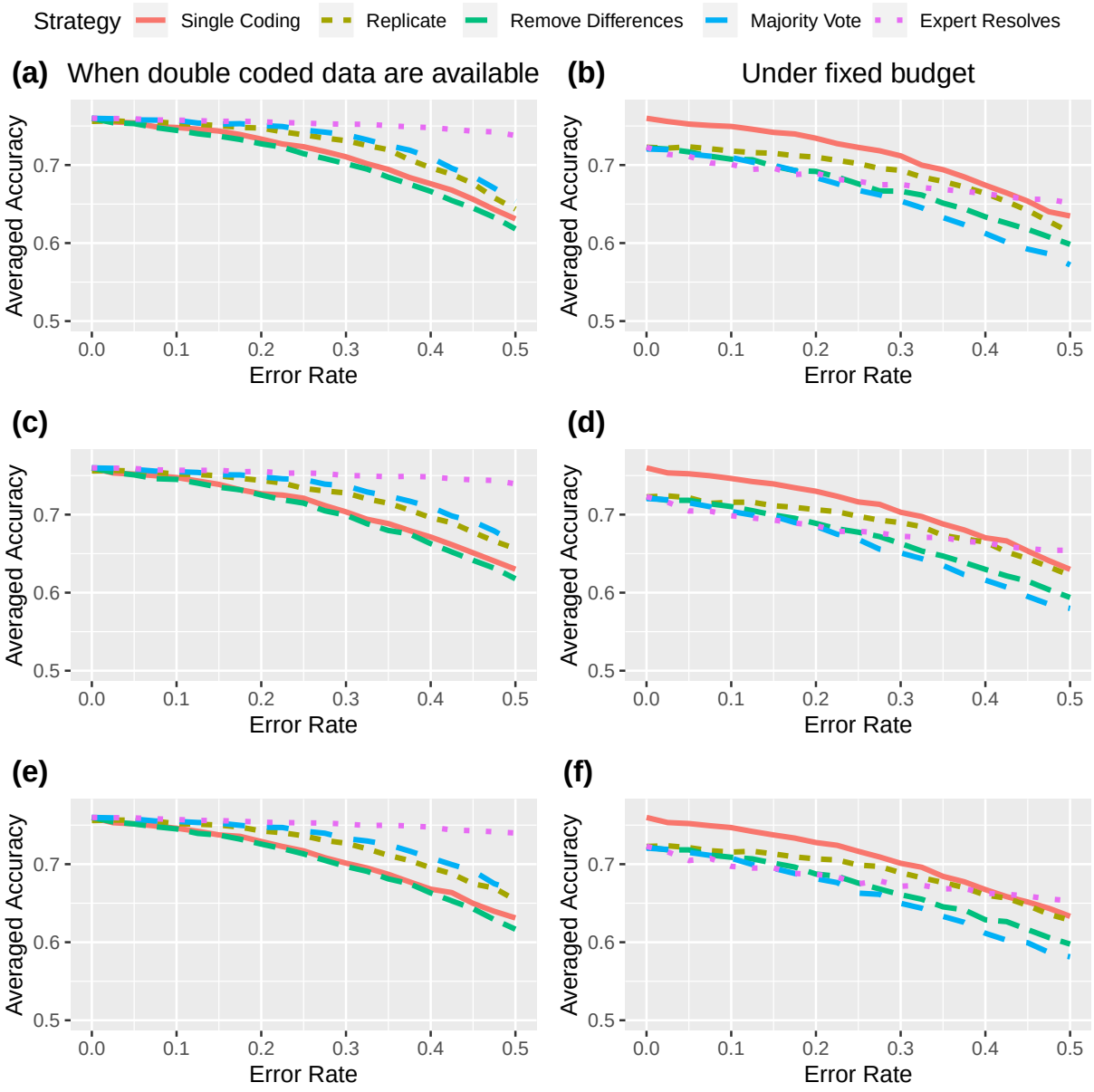


Figure 2.6: Average accuracy as a function of the error rate p in simulations using the Patient Joe data, when we assume the coding matrix is M_3. Top plots are for $g_1 = 0.2$ and $g_2 = 0.5$, middle plots are for $g_1 = 0.5$ and $g_2 = 0.2$, and bottom plots are for $g_1 = 0.5$ and $g_2 = 0.5$. The first column shows simulations when double coded data are available while the second column shows when the budget is fixed.

2.5 Applying Double Coding Strategies: Two Case Studies

In Section 2.3, we simulated the double codes assuming coders follow a known 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 double coded data sets: Happiness and Patient Joe. In both the Happiness and Patient Joe data sets, two coders coded all data independently, and the disagreement between them was resolved by an expert or a group of researchers. We can implement all strategies (except "majority vote" due to the lack of a third coder) on the two data sets based on available codes.

Same as the simulations in Section 2.3, I use SVMs. The tuning parameter C of SVMs are selected through 10-fold cross-validation; The value of C is allowed to vary for different strategies. Then, we run 10-fold cross-validation 100 times.

The Happiness data set has unordered classes while the 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, I need first to estimate the coding error rate. Since the coding error rate is unknown, we draw a random sample of 100 texts from each data set. I 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, I expect under a fixed budget single coding performs best and when double codes are already available "expert resolves" performs best. I compare all strategies (except "majority vote") to verify the expectations. The mean predictive accuracy of the 100 cross-validations is presented in Figure 2.7 and 2.8.

For the Happiness data (Figure 2.7), when double codes are available, bootstrap tests show that "expert resolves" and "replicate" improve automatic coding significantly compared with single coding ($p = 0.025$ for "expert resolves" and $p = 0.03$ 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). Although "replicate" perform better than expected, this result is consistent with the results in Section 2.3.

For the Patient Joe data (Figure 2.8), we find that "expert resolves" works best when double coded data are available and single coding works best under a fixed budget. 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 worse 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%.

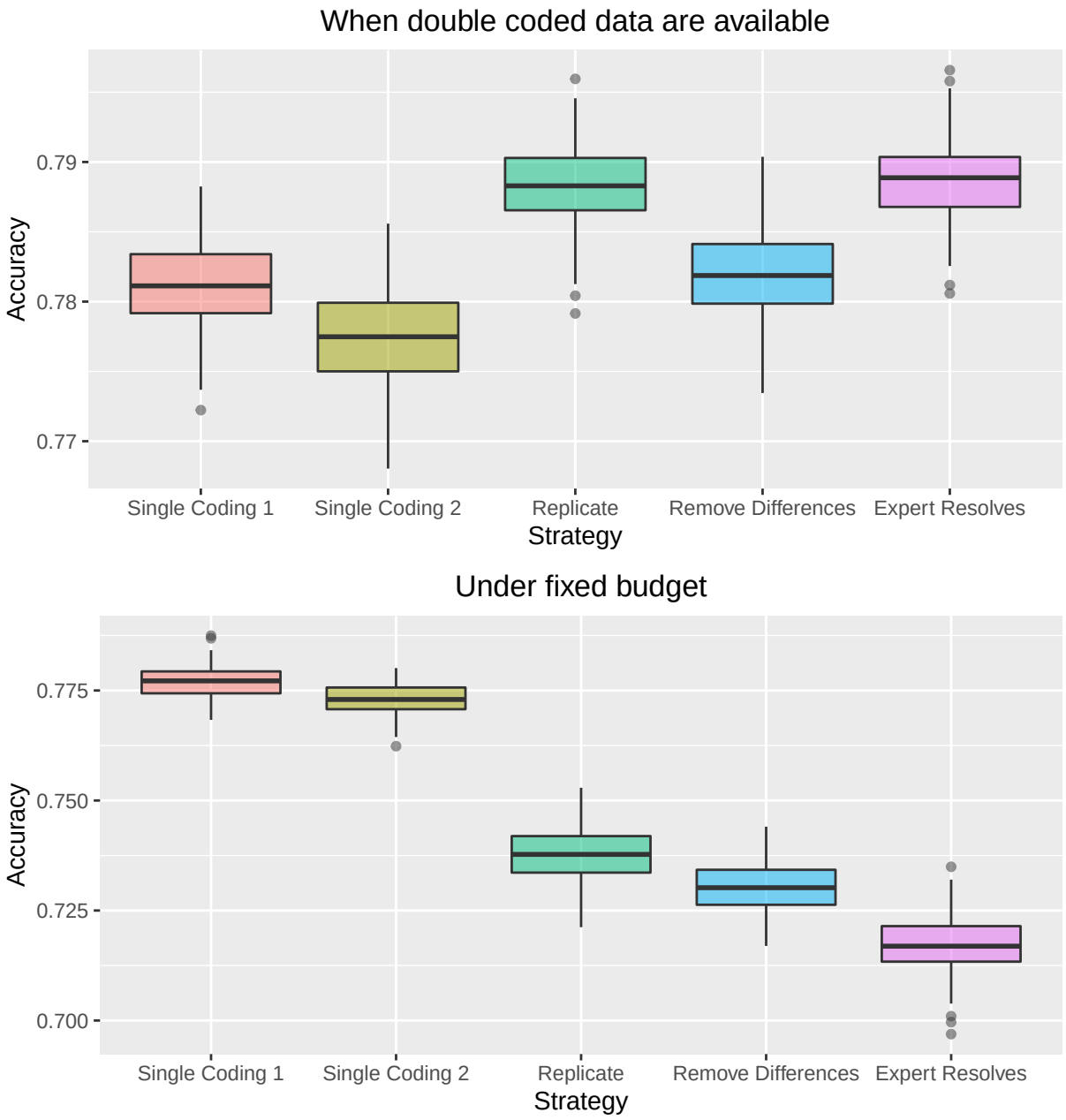


Figure 2.7: Boxplot of the predictive accuracy on the Happiness data when double codes are available (top) and under a fixed budget (bottom).

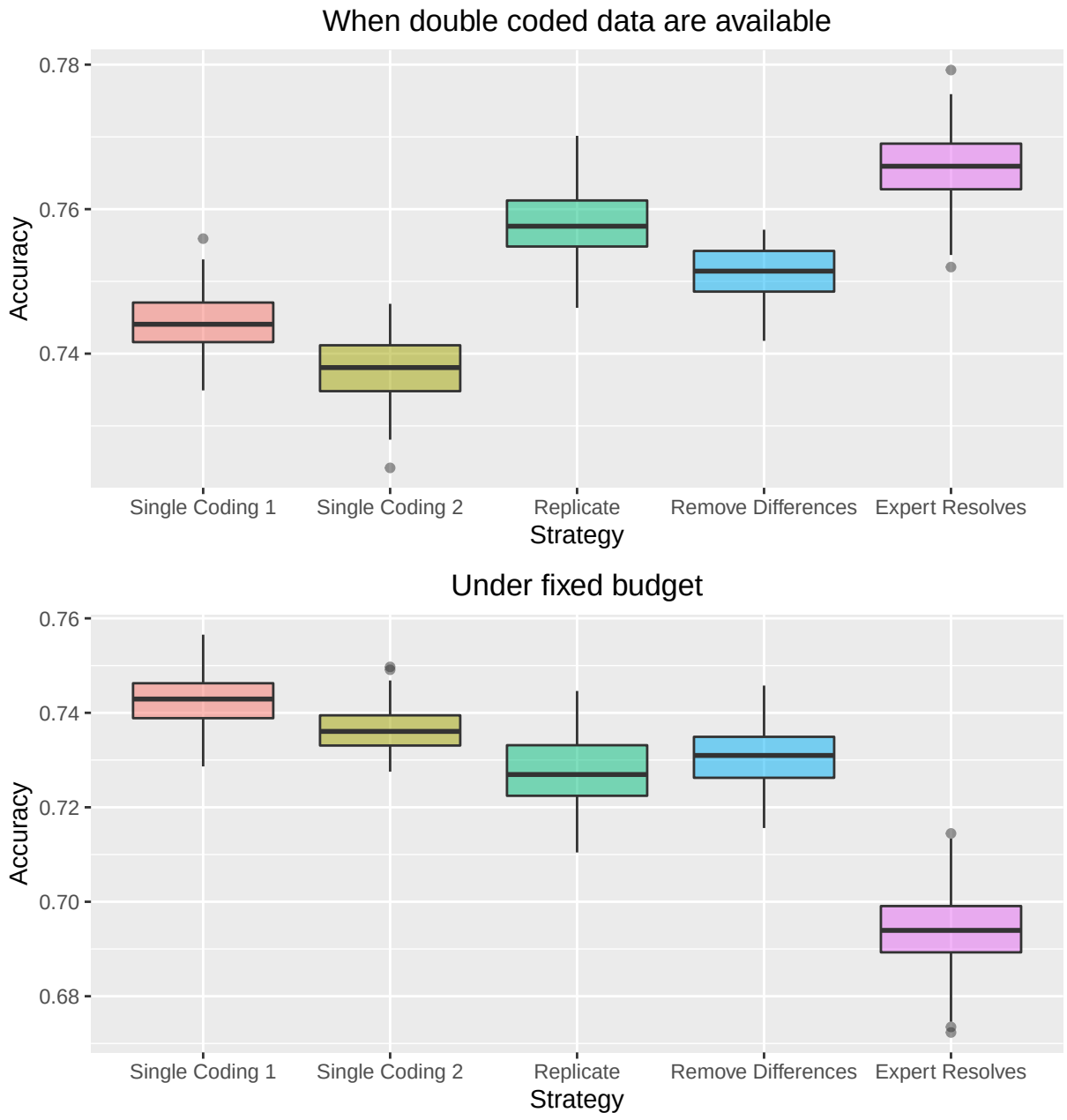


Figure 2.8: Boxplot of the predictive accuracy on the Patient Joe data when double codes are available (top) and under a fixed budget (bottom).

2.6 Comparing Single Coding and "Expert Resolves" by Resampling

As shown in Section 2.3, the threshold beyond which double coding outperforms single coding is data-dependent. In other words, without further information, we are not sure what the threshold is exactly nor whether double coding is preferable for the data of our interest. Also, in practice, it is impossible to "adjust" the coding error rate to look for the intersection as in Figure 2.1 and 2.3. Therefore, a small-sized experiment for comparing single coding and double coding is essential for researchers to decide which coding strategy to use.

Naturally, researchers can apply the two strategies to code a random subset of data, train models on the coded subset, and compare the predictions directly. The coding strategy that generates better prediction would be selected to be the coding strategy for the rest of the data. We refer such a way of experiments as direct comparison. Since errors in regular coders' coding are random, the decision made based on one run of direct comparison may be unstable. To reduce the effect of randomness, researchers may need to code the subset and run the direct comparison process multiple times. Rather than comparing two strategies directly, I propose that researchers can compare single coding and double coding by a resampling experiment, which I call "resampling comparison".

Suppose we want to compare single coding and best performed double coding strategy "expert resolves" under a fixed budget. The resampling process needs three coded subsets: Set A (of size N_1) coded by two regular coders, set B (of size N_2) coded by an expert, and another expert-coded set T (of size N_3) for testing. The basic idea of the resampling technique is to construct imitative training sets for the two coding strategies by selecting observations from the coded observations, fit models on the constructed training sets and compare predictions on set T. Taking binary classification as an example, the resampling experiment includes several steps:

1. Randomly select three non-overlapping subsets A, B and T from the dataset as observations for the experiment. The size of the three sets are N_1, N_2 and N_3 respectively. Set A is coded by two regular coders, while B and T are coded by an expert. Generally, N_1 should be larger than N_2 and N_3. The ratio of N_1 over $N_2 + N_3$ depends on the availability of the expert.

2. In order to apply the formulas in Section 2.2 to get the size of coded sets under a fixed budget, we need to estimate the coding error rate p (as it is rarely known in practice). In the binary case, we may estimate p by

$$\hat{p} = \frac{1 - \sqrt{1 - 2N_{diff}/N_1}}{2}$$

where N_{diff} is the number of differently coded observations in set A. This formula is derived by assuming regular coders make mistakes following a binomial distribution. The probability that an observation in set A is coded differently is $2p(1-p)$. Then the expected number of differently double-coded observations $E(N_{diff})$ is $2N_1p(1-p)$. For the purpose of estimation, I replace $E(N_{diff})$ with its observed value. The estimation formula holds if $N_{diff}/N_1 \leq 0.5$. If $N_{diff}/N_1 > 0.5$, except for the effect of randomness (which can be reduced by increasing the size of the coder-coded set), it may be due to that regular coders are not much better than random guessing.

3. The size of the training set for single coding is set to be n_1. For simplicity, n_1 can be the same as N_1 ($n_1 = N_1$). Then, using the estimated coding error rate, we calculate the number of training observations for "expert resolves" n_2 under fixed budget using the following formula:

$$n_2 = \frac{n_1/2}{1 + t\hat{p} - t\hat{p}^2} = \frac{N_1/2}{1 + t\hat{p} - t\hat{p}^2},$$

where t is the relative cost of an expert over an ordinary coder.

4. The training set for single coding is constructed by select n_1 observations randomly without replacement from the coder-coded set A. The classes of these training observations are randomly assigned to be one of the two codes with equal probability.

These n_1 observations are used as the training set of single coding by resampling.

5. To create the training set for "expert resolves", we sample n_2 observations from the coder-coded set A with replacement. If a selected observation is coded differently by the two regular coders, it is replaced by a randomly selected (with replacement) observation from the expert-coded set B. Then the n_2 observations are used as the training set of "expert resolves" by resampling.

6. Statistical learning models are fitted on the training sets by resampling of single coding and "expert resolves" respectively. The accuracy of their predictions on the test set T is then calculated.

7. Repeat Step 4 - 6 multiple times to reduce the effect of randomness. The coding strategy that results in better prediction is selected to be applied on the rest of the data. We have two ways to decide which coding strategy predicts better: 1) Way 1: we may compare the average of prediction accuracy from the multiple runs in the resampling experiment and select the coding strategy that results in higher average accuracy. 2) Way 2: we compare the prediction accuracy in each run and select the coding strategy that has higher accuracy on more than half of the runs.

To show the validity of the proposed procedure, we use the logit-simulated dataset. There are 2,000 observations in the data set. It contains 500 randomly generated ex-

planatory variables and a binary response. The probability of the response to be 1 is calculated using a logistic regression model. Specifically, all x-variables are indicator variables and they are drawn in two steps. First, a continuous z-variable is drawn from a normal distribution $Z \sim U(0,1)$. Second, a threshold t is drawn from a normal distribution: $t \sim N(0.1, 0.000625)$. Then $X_{ij} = I(Z_{ij} \geq t_{ij})$ where $I(.)$ is the indicator function, i denotes the observations and j the x-variables with $j = 1, 2, ..., 500$. Each indicator variable represents the presence or absence of the corresponding word in the text. Since some words are more frequent than others, the threshold is not constant.

All x-variables and 100 randomly selected pairwise interactions are used in the logistic regression model to get the probability of response equal to 1. Coefficients of the model are simulated by taking the sum of random values from $N(0, 0.16)$ and random values from $U(-1, 0.8)$. The response (label) Y is simulated by Bernoulli distribution $B(1, p_y)$, where p_y is calculated using the logit model with the generated explanatory variables and coefficients. The process of the data generation is illustrated in Figure 2.9 as well.

The simulated data are similar to real text data in several aspects: a) All x-variables are indicator variables (0/1) representing presence or absence of a word; b) The explanatory variables are sparse, i.e., only a small proportion of words are "present" (with indicator variable equal to "1") in any text; c) The number of x-variables, 500, is relatively large.

We set $N_1 = n_1 = 900$, $N_2 = 300$, and calculate the prediction accuracy on the test

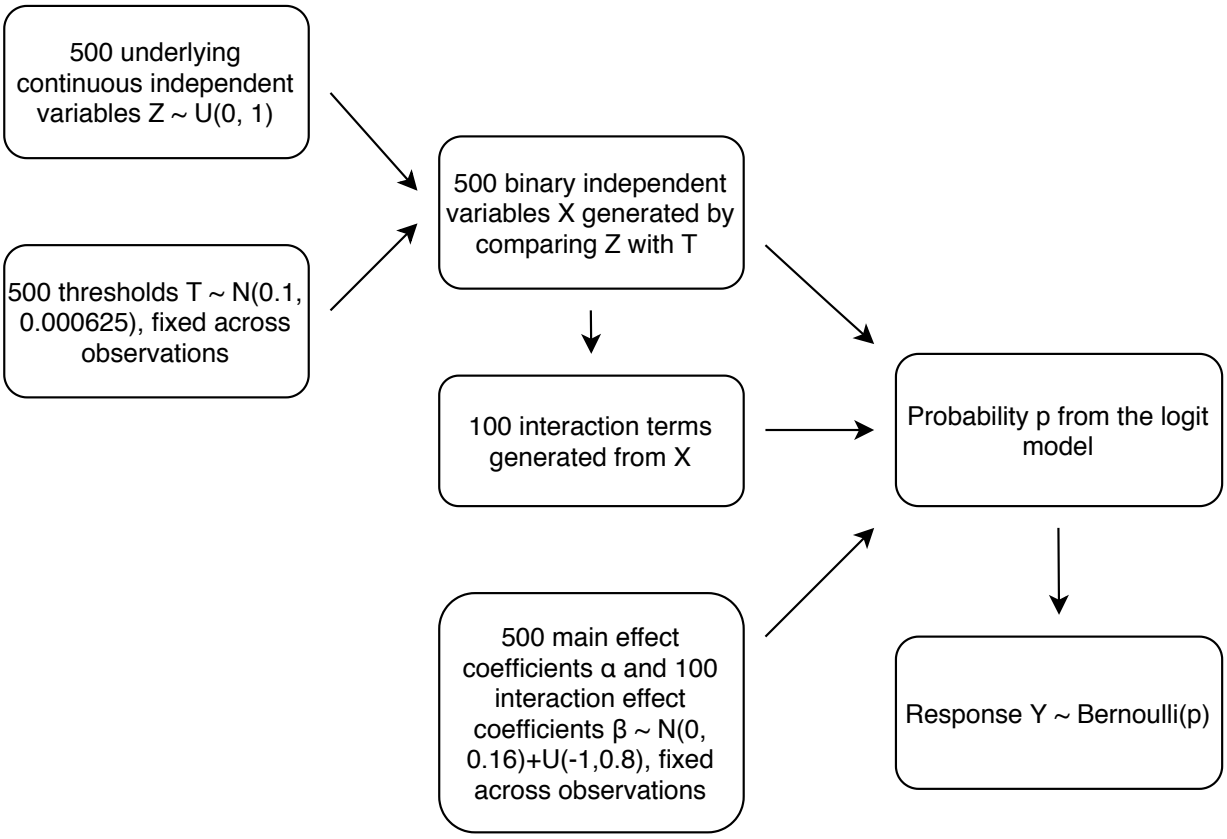


Figure 2.9: Generating process of the logit-simulated data.

set of $N_3 = 800$ observations. Same as the previous sections, the codes by regular coders were simulated from true labels with a certain probability to be wrong. The simulation of the resampling experiment is repeated 100 times, and the average prediction accuracy is shown in Figure 2.10.

If the prediction accuracy in resampling comparison is similar to the accuracy in direct comparison, the decision made based on the two approaches would be similar as well. We can see from Figure 2.10 that the prediction accuracy of "expert resolves" in resampling is similar to that of applying "expert resolves" directly, and the threshold I_1 found by the resampling process is almost identical to the threshold I_2 by direct comparison. Thus, in expectation, the proposed resampling comparison would suggest almost the same coding strategy as direct comparison. For coding error rates between 0 and 0.5 except for the small interval between I_1 and I_2, if "expert resolves" outperforms single coding in the proposed resampling comparison, it would also be selected in direct comparison, and vice versa. For coding error rates between I_1 and I_2, resampling comparison and direct comparison give different suggestions, yet single coding and "expert resolves" perform similarly so that a wrong decision is not a big issue.

In practice, researchers are unlikely to do a comparison experiment on the whole data set. Instead, the comparison is often applied on a much smaller set. To show that the performance of resampling comparison is no worse (or even better) than direct comparison

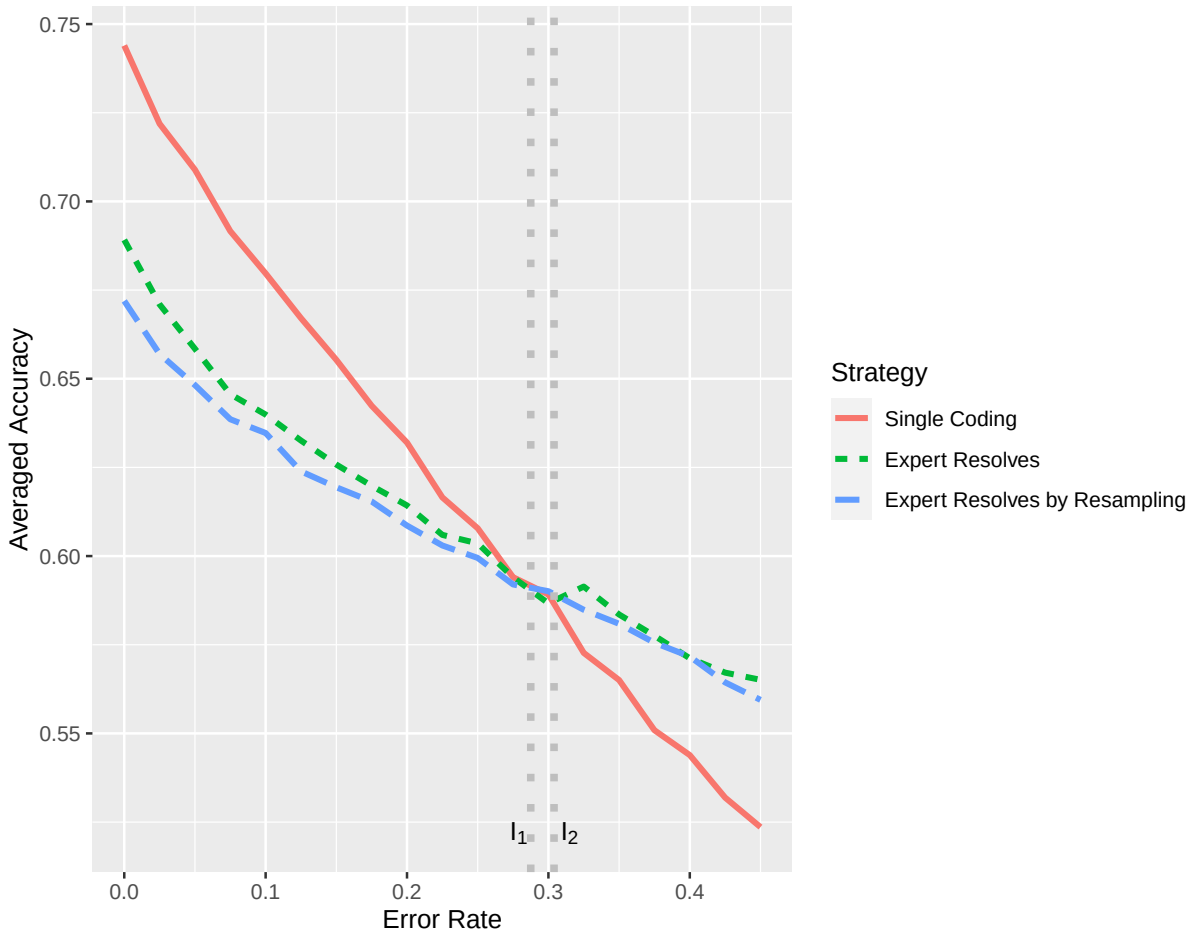


Figure 2.10: Average prediction accuracy of single coding and "expert resolves" in the proposed comparison by resampling and in direct comparison. I_1 is the error rate corresponding to the interaction between single coding and "expert resolves", and I_2 is the error rate corresponding the the interaction between single coding and "expert resolves" by resampling.

when only a small proportion of data are coded, I run both comparison methods on a randomly selected subset of the logit-simulated data. I re-set $N_1 = n_1 = 300$, $N_2 = 100$ and $N_3 = 200$, so that each of the regular coders and the expert code 300 observations. As experiments on a small subset have no guarantee that the selected strategy is optimal, I evaluate the two comparison techniques in terms of the percentage of times (in 100 repeated simulations of the experiment) that a correct selection (which is to use sing coding when the error rate is less than 30% and "expert resolves" otherwise) is made.

Figure 2.11 shows how direct and resampling comparison work in simulations of the small-sized experiment. For example, when the error rate is 0.35, in about 54% of times direct comparison selects the right coding strategy while resampling comparison has about 65% using way 1 and 57% using way 2. We find that small-sized experiments of direct and resampling comparison have similar curvature of performance: When the error rate is low ($< 15\%$), both comparison approaches have a high probability ($> 75\%$) of selecting the right coding strategy (single coding); When the error rate is high ($> 35\%$), both approaches have more than half of the chances of making a correct decision ("expert resolves"); When the error rate is close to the threshold of single coding vs. "expert resolves" (which is about 30%), a small-sized experiment does not help much.

Despite the similar performance curve, the proposed resampling technique has a higher chance of selecting the better coding strategy on average. Over the error rate range $[0, 0.45]$,

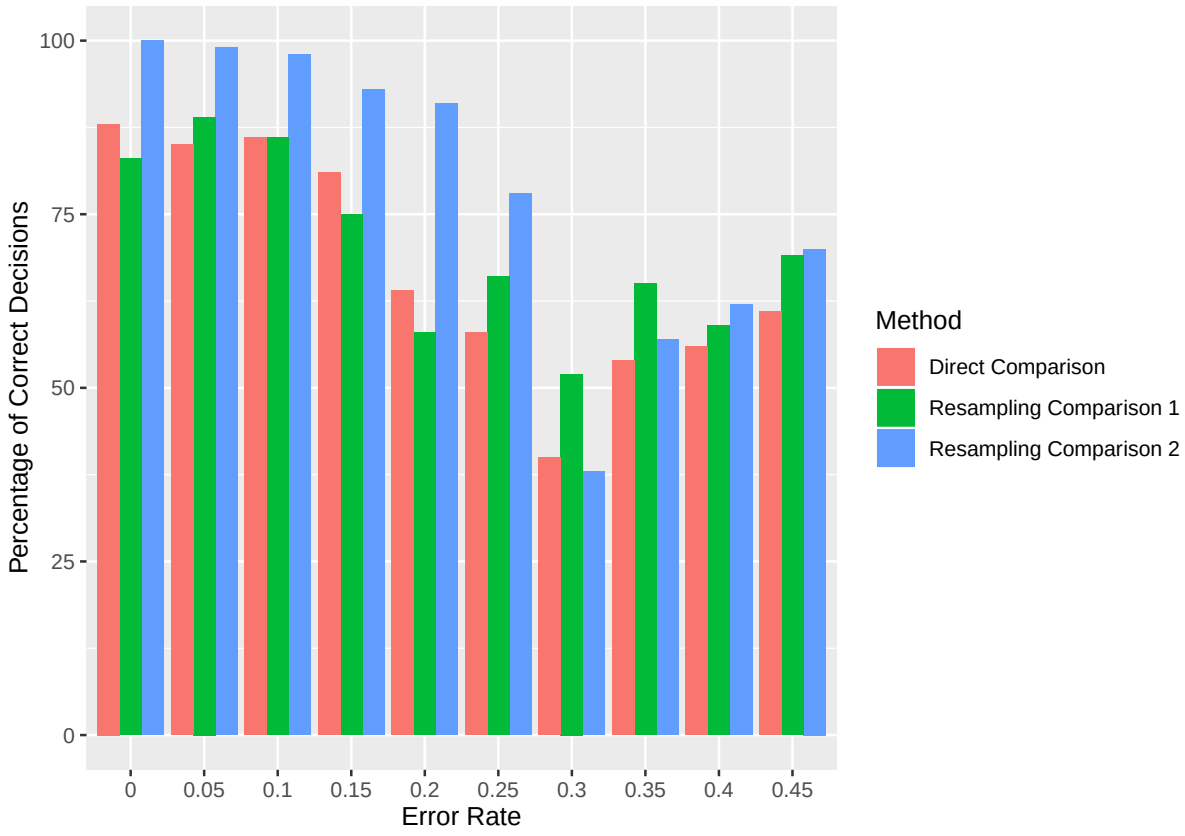


Figure 2.11: Percentage of correct decisions in 100 repeated simulations based on direct comparison and resampling comparison when applied on subsets of the training set. $N_1 = n_1 = 300$, $N_2 = 100$ and $N_3 = 200$ in resampling comparison, which involves 20 runs of resampling. Resampling comparison 1 refers to the first way of decision making (compare the average accuracy across runs), and resampling comparison 2 refers to the second (compare the number of runs that accuracy is higher).

resampling comparison has 71.7% and 78.7% (in way 1 and 2 respectively) probability of making the right decision, yet direct comparison has 68.2%.

In addition to the higher chance of making the right selection, the proposed comparison method has the following advantages: 1) After manual coding for one time, we can repeat the comparisons multiple times by taking different "samples with replacement". This allows, with a small cost on manual coding, a reduction in the effect of randomness, which is a potential problem if the comparison is done only once; 2) The coding work of the regular coders and expert can be conducted simultaneously, while in direct comparison, the expert must wait until regular coders finish their work.

2.7 Discussion

I have explored whether and how double coding can be used to improve automatic classification of responses to open-ended questions. Five strategies are proposed for resolving potential inter-coder disagreement in double coding. I 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. In the simulations, the threshold error rate is around $20 \sim 35\%$ for binary classification and $35 \sim 45\%$ for

multi-class classification. This suggests that, when there are only two classes, researchers may use single coding if they think regular coders have coding accuracy over 80%, or apply "expert resolves" double-coding strategy if coding accuracy is less than 65%. When there are multiple classes, single coding seems to be a practical choice: if the coding error rate exceeds 45%, researchers need to modify the codebook or redesign the coding procedure to reduce coding error. Further, when double coding is preferable, "expert resolves" is the best strategy. 2) When texts have already been double coded, I 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 or to have a third coder to vote.

It is somewhat surprising that removing inter-coder disagreement beats or works similarly as the "majority vote" strategy that involves 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 remains as both coders may have miscoded, the probability of which is $(1-p)^2$ in binary classification. I conclude that you would rather have a small but clean data set than a large but messy one. Of course, when generalizing to more than two outcome classes, whether "remove differences" would still beat "majority vote" needs to be discussed case by case.

Although not literally true, it may be a reasonable approximation to assume an expert

is always correct in the model, considering the fact that 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 allow 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, I reran the simulations assuming the expert's coding error rate is one tenth of that of a regular coder (not shown). The threshold at which the strategy "expert resolves" is preferable over single coding become slightly larger (by about 2%), meaning that single coding remains attractive at slightly higher coding inaccuracies. Otherwise, results are consistent with the previous results.

Coding errors can be due to human errors or due to ambiguity of the text. (I 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 simulations have focused on 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 the hope 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.

This chapter focuses on the coding procedure in classifying short texts such as open-ended responses. D'Orazio et al. (2016) have taken a different approach to reducing the cost of manually coding all texts. Rather than employing statistical learning, they recruited coders on 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 even moderate) and statistical learning models may be unstable for small sample sizes; 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 n-gram approach to statistical learning does not tend to work well on long texts.

The limitations of this study include: 1) Although I identify the existence of a threshold between single coding and double coding, researchers do not know what its value is for a specific data set. Therefore, I propose a resampling technique allowing researchers to estimate the threshold based on a small-sized experiment. 2) In the simulations, I only use

SVM when training 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 also be used. 3) I assume that regular coders have the same coding matrix. This is perhaps an oversimplified assumption. However, assuming different coding matrices would further increase complexity and I have no reason to believe that it would make a difference in the conclusions. Also, the coding errors of the same coder may be correlated as a coder is likely to make the same mistakes repeatedly. So my simulations may even underestimate the performance of double coding, because a second opinion may counterbalance the bias of a particular coder.

In summary, when human coding 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 may remove differently coded data from the training data - even though this reduces the size of the training set - or employ a third coder to resolve the differences.

Chapter 3

A Model-assisted Approach for Finding Coding Errors in Manual Coding of Open-ended Questions

3.1 Introduction

Text answers are awkward for quantitative analysis. Usually, text answers are coded manually into one of several categories as specified by a coding manual. To judge the quality of the coding process, a *random* subset of answers is double coded and the intercoder reliabil-

ity Kappa is computed. If intercoder reliability is low, it indicates that the coding quality is poor and the coding process should be improved. For example, one might re-work on the coding manual, re-define or combine codes, or re-train the coders.

Also, we can improve coding quality by identifying and correcting coding errors. Double coding the whole data set will detect most coding errors; the probability of both coders coding the same text answer incorrectly using the same incorrect code is small. However, for cost reasons, typically only a subset of the text answers is double coded. Even when the intercoder reliability is acceptable, errors remain in the majority of the text answers that are single coded.

If we can identify single-coded text answers whose codes are suspicious in light of the double-coded subset, we could check the suspicious codes to improve the coding quality. Focusing on suspicious codes may be a worthwhile compromise between double coding the entire data (at high cost) and making no attempt at reducing coding errors in the single-coded data.

In this chapter, I propose a model-assisted approach to find suspicious codes in single-coded data while retaining the ability to assess reliability. The outline of this chapter is as follows: Section 3.2 introduces the proposed process for identifying suspicious codes. Section 3.3 contains case studies to evaluate the proposed approach to identifying suspicious codes. Section 3.4 concludes with a discussion.

3.2 Methodology

My proposed method for finding suspicious codes needs to ensure that the original purpose of double coding – assessing intercoder reliability Kappa – is not compromised: Computing Kappa requires that the double-coded subset is selected at random. The basic idea is to select a double-coded subset in two steps: the first step selects observations randomly to compute Kappa, and the second step selects observations with a high risk of error.

Therefore, I consider the following situation: There are N observations. A random subset of the data (size N_1) is double coded to compute Kappa. The remainder of the data, $N - N_1$ observations, are single coded. I wish to identify suspicious codes that are likely coding errors among the single-coded observations. A coding error occurs when a coder's code does not match the gold standard code.

Broadly, I use statistical learning models to estimate the probability of disagreement: the probability of a single-coded observation would lead to a coding disagreement if it was double coded. In practice, it is unlikely that two coders make the same mistake (and even if they do, it is very difficult to find out). So different codes from the two coders mean at least one of the coders is wrong. The probability of disagreement indicates a risk of a coding error. The top N_2 observations with the highest probability of disagreement are then also double coded (they are already single coded; a second coder is added). If the

double coding of the N_2 observations leads to different codes, this is resolved (e.g. by employing an expert or a group discussion of coders) to get gold standard codes.

I propose two ways to predict the risk of intercoder disagreement: we can either predict the codes of the second coder and compare with the first coder's codes to find disagreements, or predict the disagreement directly. I call the coder who codes the whole data set coder 1, and the coder who codes only the first N_1 observations coder 2. The two approaches are detailed as follows:

- Predict Codes First: Train the model on coder 2's codes in the first N_1 observations. For each observation in the single-coded data, predict the probability that coder 2 gets the same code as coder 1. The risk of disagreement is 1 minus this probability.

- Predict Disagreement Directly: Train the model on whether coder 1 and coder 2 disagree in the first N_1 observations, using coder 1's codes as one of the explanatory variables in the model. For each observation in the single-coded data, predict the probability that coder 1 and coder 2 disagree. The risk of disagreement is the associated probability.

I note some details: For the method Predict Disagreement Directly, I find that including coder 1's codes as a covariate improved prediction accuracy. Some codes may have a higher risk of disagreement. Including coder 1's code in the model enables the model to explicitly

model that. (Coder 2's codes are only available for the N_1 double-coded observations and therefore cannot be used for prediction.) By contrast, for the method Predict Codes First, I find that including coder 1's code does not improve prediction accuracy.

The proposed method can identify suspicious codes, but not all suspicious codes will turn out to be incorrect. I therefore evaluate the proposed methods relative to "Random Selection" as the baseline method.

- Random Selection: assume the risk of disagreement is constant for the $N - N_1$ single-coded observations. So select N_2 observations randomly from the single-coded observations and get them coded by coder 2. This is equivalent to randomly select $N_1 + N_2$ observations for double coding in the first place.

3.3 Case Studies on Double-coded Data

I evaluated the three strategies, "Predict Codes First", "Predict Disagreement Directly" and "Random Selection", on the Patient Joe, Happiness and Democracy data sets. The proposed methods require the use of a model for prediction. The model must accommodate multi-class outcomes and be able to cope with a large number of variables and multi-collinearity. In the case studies, I choose support vector machines (SVMs) with a linear kernel.

To evaluate intercoder disagreement, typically 250 double-coded observations are sufficient. Therefore, I choose $N_1 = 250$ for the case studies. In addition, I then identify $N_2 = 250$ suspicious text answers. For "Predict Codes First" and "Predict Disagreement Directly", I select the 250 answers with the highest model-based risk of disagreement. For "Random Selection" I select random answers. To learn about the variability of the process, I run the case studies 1,000 times.

Figure 3.1 shows boxplots of the number of disagreements found in the $N_2 = 250$ additional double-coded answers using the three coding methods. On average, the method "Predict Codes First" finds the most disagreements for all three data sets. For the Patient Joe and Democracy data, this method identifies more than twice as many disagreements as compared to "Random Selection"; for the Happiness data about three times as many. For the Patient Joe data, I find about 150 of 250 suspicious codes correspond to intercoder disagreements. The number of disagreements I find is lower in other data sets because the total number of disagreements in the data is lower. As a percentage of the total number of disagreements, using "Predict Codes First" I find 36.1%, 50.6% and 46.2% (on average) of total disagreements in the Patient Joe, Happiness and Democracy data sets, respectively. The baseline method, "Random Selection", only finds 14.3%, 16.9% and 22.8% of the disagreements.

More generally, Figure 3.2 shows the mean performance of the three methods as a

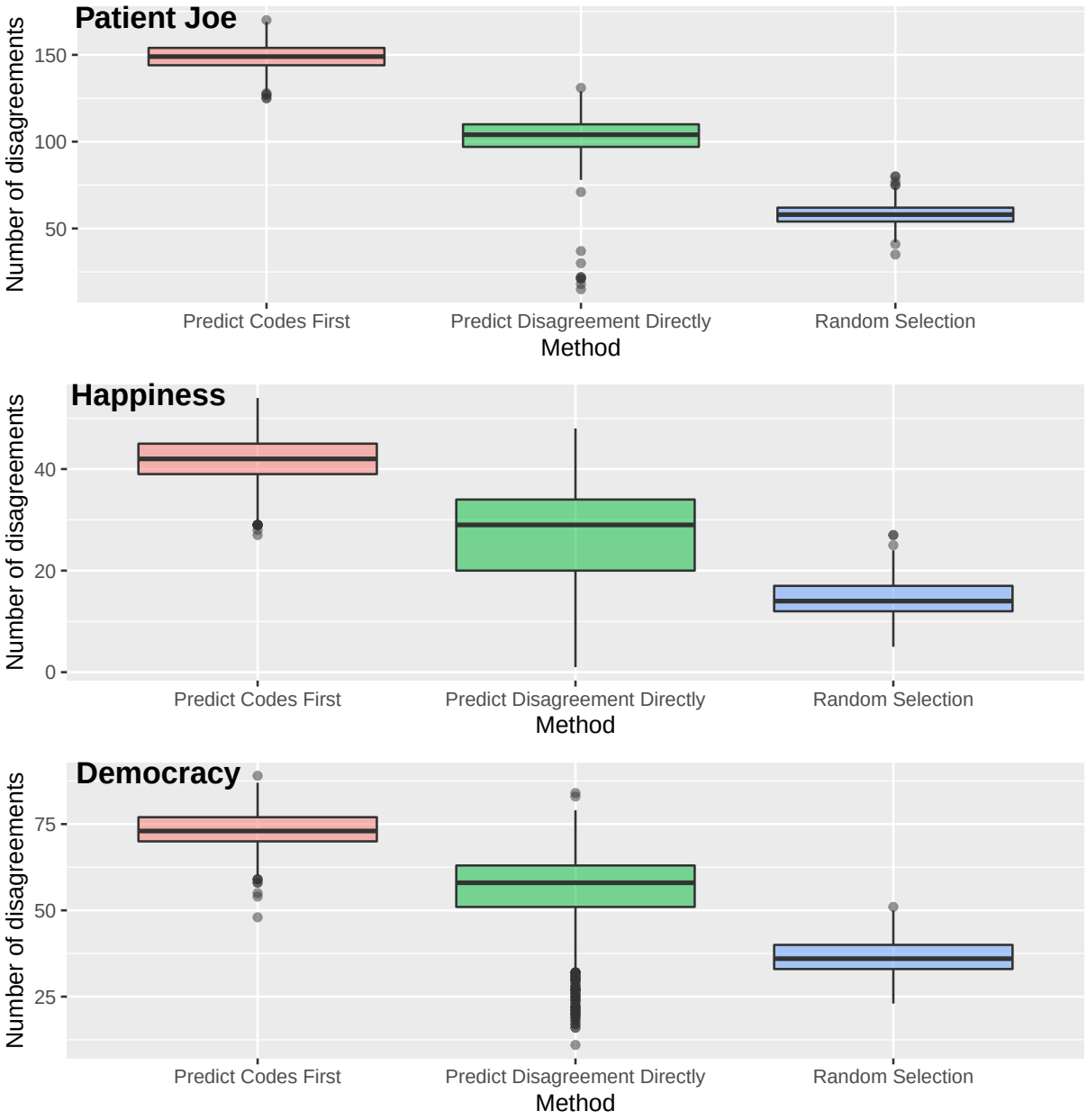


Figure 3.1: Boxplot of the number of intercoder disagreements found in the additional 250 double-coded answers for the Patient Joe, Happiness and Democracy data sets.

function of N_2. The plot also contains 95% empirical confidence intervals.

All the three plots in Figure 3.2 tell similar stories: "Predict Code First" outperforms "Predict Disagreement Directly" and "Random Selection". This is consistent with what I find in Figure 3.1. Note that for the Happiness and Democracy data at low values for N_2 two curves cross but there is no statistically significant difference based on the confidence intervals.

3.4 Discussion

I have introduced a model-assisted procedure for finding coding errors in single-coded answers to open-ended questions. Starting with a double-coded *random* sample — which enables computing Kappa to assess intercoder reliability — I use the model to find the most suspicious codes among the single-coded data. My finding is that the method "Predict Codes First" finds two to three times as many coding errors as compared to random guessing when selecting an additional 250 text answers for double coding.

I have answered the following question: If our budget allows double coding another N_2 text answers, which observations should we choose to improve coding quality? More broadly, does the improved coding quality justify the extra budget for additional double coding? Figure 3.2 shows the diminishing rates of return. When the number of disagree-

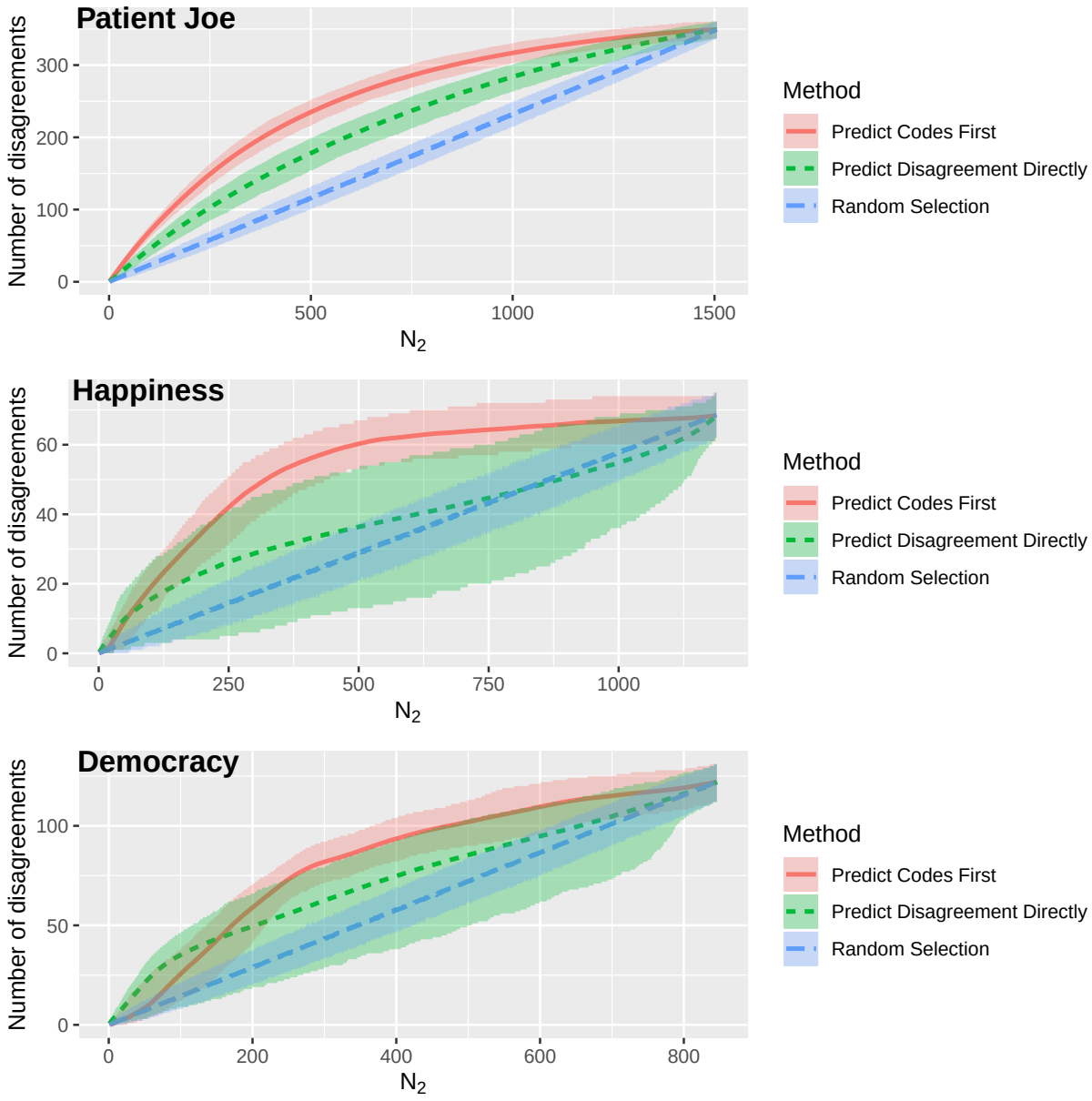


Figure 3.2: The number of intercoder disagreements as a function of additional N_2 double-coded answers by the three methods for the Patient Joe, Happiness and Democracy data sets. The disagreements in the initial N_1 double-coded answers are not shown in the graphs.

ments is small, the Happiness and Democracy data show double coding an additional 250 answers will identify about 40 (Happiness) and 70 disagreements (Democracy), on average. When the number of disagreements is large, the Patient Joe data show double coding an additional 250 answers will identify 150 disagreements, on average, and the next 250 answers will identify another 100 disagreements. This appears to be an attractive proposition.

The size of the random subset, N_1, has to be chosen large enough to compute Kappa reliably. In the case studies, I use $N_1 = 250$ observations to compute Kappa. If N_1 were much smaller than N_2, one could employ a 2-step procedure: first, identify the most suspicious $N_2/2$ codes; second, train the model on all $N_1 + N_2/2$ double-coded observations and identify the most suspicious remaining $N_2/2$ codes.

I have compared two approaches for predicting the risk of disagreement: predict the codes first and compute the probability of the code chosen by the single-coder, or predict disagreement directly. The case studies show predicting the codes first performs better. My intuition is as follows: Predicting disagreement is a binary prediction whereas predicting codes is a multi-class prediction. It must be easier to predict individual codes accurately than to predict disagreements across all codes combined.

The limitations of the study include: 1) I use SVMs in the case studies. Other off-the-shelf statistical learning methods (such as gradient boosting and random forests) could be used instead. In my experience results are robust with respect to the choice of models. 2) A

random subset of double-coded observations is needed to calculate Kappa. The additional N_2 observations are not selected at random and cannot be used to increase the subset on which Kappa is computed. As long as N_1 is not too small, I believe the implied tradeoff between a larger subset to compute Kappa and a greater coding quality is well worth it.

In summary, I proposed a model-assisted procedure to identify single-coded observations with a high risk of a coding error. I conclude that if the budget allows additional double coding, then this procedure is the method of choice to improve coding quality. The greater the intercoder disagreement, the greater the benefit.

Chapter 4

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

4.1 Introduction

Both human and automatic coding make mistakes but for different reasons. Manual coding error stems from human error, ambiguous text answers, and maybe 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 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 which 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.

This chapter explores whether and to what extent human coders and automated coders make similar coding mistakes. The outline is as follows: Section 4.2 investigates similarities and differences between human and automatic coding. Section 4.3 discusses conclusions and limitations.

4.2 Comparison between Manual Coding and Automatic Coding using Examples

In this chapter, I use three double-coded datasets: the Patient Joe, Happiness and Democracy data sets. I also use support vector machines (SVMs) and random forests (RF) as representatives of statistical learning models (James et al., 2013). I randomly split each of the three datasets into a training set and a test set (as specified in Table 1.1). The SVM and random forests models are trained on the "gold standard coding" (the coding after disagreement-resolution) of the training data. The trained models are then used to predict the codes of the test data. These predicted codes are referred to as the codes of automated coders in later sections.

4.2.1 Do Automated Coders Achieve Similar Coding Accuracy as Human Coders?

Figure 4.1 shows the coding accuracy of the two automated coders and two human coders in the three datasets. The coding accuracy is the proportion of codes that match the gold standard codes. 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 training on the gold standard codes, the coding error of automated coders is only due to statistical generalization error, not due to human error. The coding accuracy (shown in Figure 4.1) is evaluated on the test data, as is appropriate for statistical learning models.

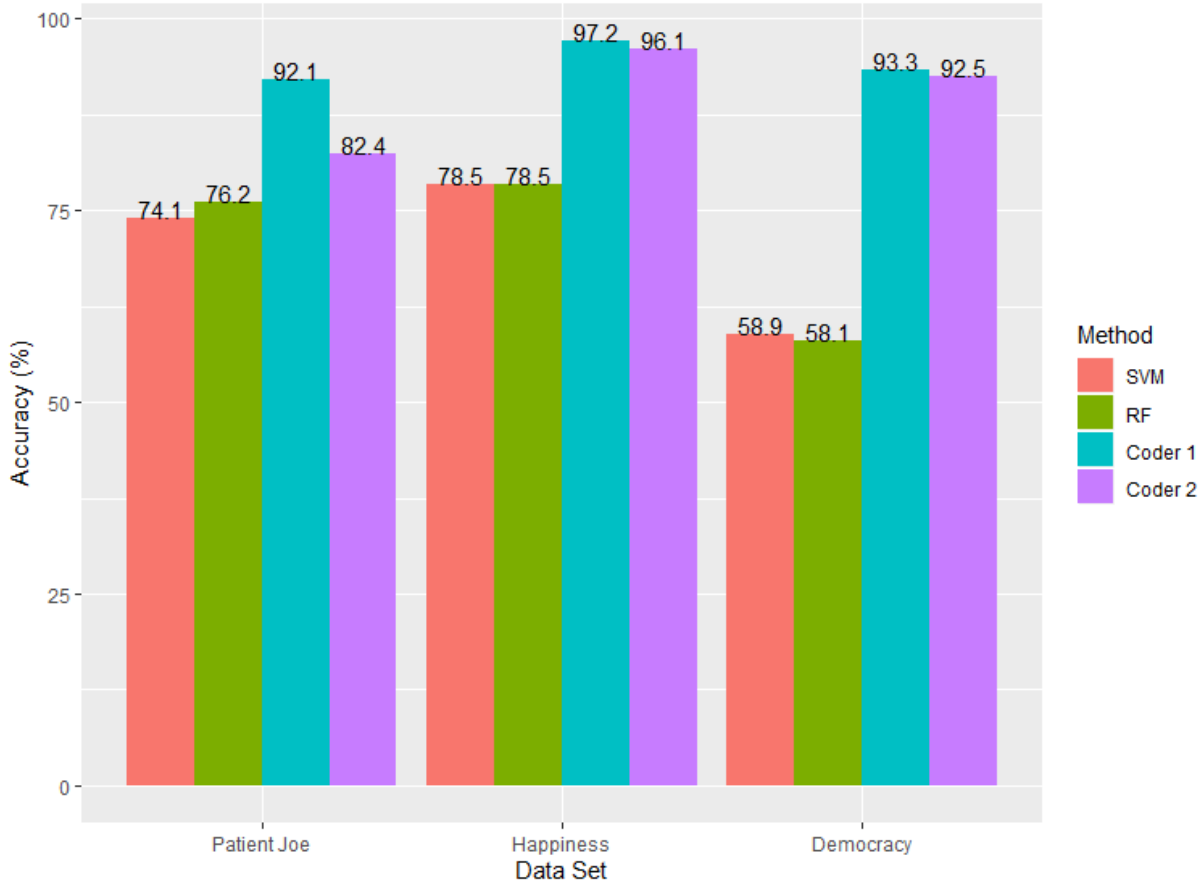


Figure 4.1: Coding accuracy of automated coders and human coders on the test data for the Patient Joe, Happiness and Democracy datasets.

We can see from Figure 4.1 that the coding accuracy of SVM and RF is lower than that

of human coders. The difference between any pair of an automated coder and a human coder is statistically significant in two-proportion z-test: the significance level is < 0.01 after Bonferroni correction. Therefore, when we investigate whether models and humans make the same mistakes, we have to remove the effect of different error rates.

4.2.2 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, it infers 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 an observation 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 or incorrect. The 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 in each subset. Rather than forming the subsets at random, I order the observations by the average of the estimated model-based coding error probability. For example, if 10 subsets are desired, each decile of the observations ordered by the mean of their automatic coding error probability forms one subset. In this chapter, I divided the test set into 36 subsets for the Patient Joe data set, 29 subsets for the Happiness data set, and 31 subsets for the Democracy data set.

Next, I compute two-way correlations among the estimated probabilities for the four coders (two automated coders and two human coders) for each data set. Since the estimated coding error probabilities for humans only exist at the aggregated level, I also estimate the coding error probabilities for automated coders in each subset to make sure the probabilities of different coders are comparable. Table 4.1 shows the correlation matrices of the estimated coding error probabilities.

I 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 is very similar as compared to the agreement between two human coders. However, the correlations only imply a tendency to find the same observations

Patient Joe

	SVM	RF	Coder 1	Coder 2
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	RF	Coder 1	Coder 2
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	RF	Coder 1	Coder 2
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

Table 4.1: Correlation matrix of estimated error probabilities for each dataset.

difficult; it is not clear whether the two models and the two humans are equally accurate or whether there are large differences in accuracy. I have already found in Section 4.2.1 that human coders are more accurate as compared to automated coders.

I 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 4.1), it does not matter which statistical learning model we choose: they are functionally equivalent. This

is different for the two human coders who have a more moderate positive correlation.

The analysis on the correlation matrices reveals pairwise similarities for the four coders, yet the overall similarities or differences of the four coders are unclear. To answer this question, I 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 mean removes the different error rates of these coders. The correlations between the coding error probabilities of each coder and the principal components are listed in Table 4.2.

The three analyses for the three datasets tell similar stories. The first principal component explains most of the variation ($65 \sim 80\%$) 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. Another principal component (the second component for the Patient Joe and the third component for the Happiness and Democracy) represents specific 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 commonalities than differences.

4.2.3 Examples on Which Automated Coders and Human Coders Agree or Disagree

In an effort to gain further insight into the differences and similarities between human coding and automatic coding, I now look at some specific coding examples for one of the datasets, the Patient Joe data. The responses I discuss below are summarized in Table 4.3 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, and so do human coders.

The texts in my analysis are represented by n-gram variables, specifically of indicator variables of the presence or absence of single words or bigrams. As a consequence, if individual n-gram variables are highly indicative of a code (or class), automated coders 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 and random forests model are 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-grams 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 realize this response is not proactive, the two automated coders still classify it as proactive because they relied on the words "2 weeks" too heavily.

I 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 text answers that contain only new words that are not observed in the training data. Automated coders, once trained, assign these responses to a code based on the length of the responses and the absence of all known words. In the case studies, the default code of SVM and random forests in the Patient Joe data 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 completely 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 "passive". By contrast, the human coders correctly classified the response to the code "counterproductive".

4.3 Discussion

I have investigated the relationship between automatic coding and manual coding by examining the similarities and differences between their estimated coding errors. Crucially, I am able to estimate human coding error probabilities by aggregating the text answers to subsets. I find 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 I 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 bigrams): 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 and Couper, 2016), which codes

easy-to-code observations automatically and the remainder manually, is thus useful.

Limitations of this study include: 1) I use SVM and random forests as representatives of automated coders. There are other statistical learning models. I 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) I estimate the error probability of human coders by dividing the data into multiple subsets and estimating the error probability in each subset. The estimation depends on how I divide the data into subsets. I order 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 mean in each subset.

In summary, automated coders and human coders tend to find the same text answers difficult to code. While it may be useful to employ two human coders to investigate coding differences, there appears to be no point in having more than one automated coder: they make the same mistakes.

Patient Joe

	Dim 1	Dim 2	Dim 3	Dim 4
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

	Dim 1	Dim 2	Dim 3	Dim 4
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

	Dim 1	Dim 2	Dim 3	Dim 4
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%

Table 4.2: Correlation between principal components and the original estimated error probabilities. The percentage of variation explained for each principle component is also given.

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 al- leen 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

Table 4.3: Example responses for various human vs. automatic coding results in the Patient Joe data and a brief explanation about the type of response. I show both the original responses in Dutch and the English translations (using Google Translate).

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.

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

Behr, D., Bandilla, W., Kaczmirek, L., and Braun, M. (2014). Cognitive probes in web surveys: on the effect of different text box size and probing exposure on response quality. *Social Science Computer Review*, 32(4):524–533.

Behr, D., Kaczmirek, L., Bandilla, W., and 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.

Belloni, M., Brugiavini, A., Meschi, E., and Tijdens, K. (2016). Measuring and detecting

errors in occupational coding: an analysis of share data. *Journal of Official Statistics*, 32(4):917–945.

Bengston, D. N., Asah, S. T., and 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.

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.

Brennan, M. and Holdershaw, J. (1999). The effect of question tone and form on responses to open-ended questions: further data. *Marketing Bulletin - Department of Marketing, Massey University*, 10:57–64.

Bullington, J., Endres, I., and Rahman, M. (2007). Open-ended question classification using support vector machines. In *MAICS 2007*, Chicago, USA.

Büttcher, S., Clarke, C. L., and Cormack, G. V. (2016). *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, Cambridge, Massachusetts, USA.

Bylander, T. (2002). Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine Learning*, 48(1-3):287–297.

Carley, K. (1993). Coding choices for textual analysis: a comparison of content analysis and map analysis. In Marsden, P., editor, *Sociological Methodology*, volume 23, pages 75–126. Oxford, Blackwell, UK.

Chai, C. P. (2019). Text mining in survey data. *Survey Practice*, 12(1):1–14.

Chinh, B., Zade, H., Ganji, A., and Aragon, C. (2019). Ways of qualitative coding: a case study of four strategies for resolving disagreements. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, Glasgow, UK.

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

Conrad, F. G., Gagnon-Bartsch, J. A., Ferg, R. A., Schober, M. F., Pasek, J., and Hou, E. (2019). Social media as an alternative to surveys of opinions about the economy. *Social Science Computer Review*. published online first at Sep 26, 2019. `https://doi.org/10.1177/0894439319875692`.

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.

Crittenden, K. S. and 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., and 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. https://doi.org/10.1787/18151981.

Engwall, L. (1983). Research note: Linguistic analysis of an open-ended questionnaire in an organizational study. *Organization Studies*, 4(3):261–270.

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

Fleiss, J. L., Levin, B., and Paik, M. C. (2013). *Statistical Methods for Rates and Proportions*. John Wiley & Sons, New York, USA, 3rd edition.

Fowler Jr, F. J. and Fowler, F. J. (1995). *Improving Survey Questions: Design and Evaluation*. Sage, London, UK.

Fuhr, N. (1989). Models for retrieval with probabilistic indexing. *Information Processing & Management*, 25(1):55–72.

Funkhouser, G. R. and Parker, E. B. (1968). Analyzing coding reliability: the random-systematic-error coefficient. *The Public Opinion Quarterly*, 32(1):122–128.

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

Gendall, P., Menelaou, H., and Brennan, M. (1996). Open-ended questions: Some implications for mail survey research. *Marketing Bulletin - Department of Marketing, Massey University*, 7:1–8.

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

Griffith, L., Cook, D. J., Guyatt, G. H., and Charles, C. A. (1999). Comparison of open and closed questionnaire formats in obtaining demographic information from canadian general internists. *Journal of Clinical Epidemiology*, 52(10):997–1005.

Grimmer, J. and 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 meth-

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

He, Z. and Schonlau, M. (2020a). Automatic coding of text answers to open-ended questions: Should you double code the training data? *Social Science Computer Review*, 38(6):754–765.

He, Z. and Schonlau, M. (2020b). Automatic coding of open-ended questions into multiple classes: whether and how to use double coded data. *Survey Research Methods*, 14(3):267–287.

He, Z. and Schonlau, M. (N.D.). A model-assisted approach for finding coding errors in manual coding of open-ended questions. *Journal of Survey Statistics and Methodology*, submitted.

He, Z. and Schonlau, M. (to appear). Coding text answers to open-ended questions: Do human coders and statistical learning algorithms make similar mistakes? *Methods, Data, Analyses*, forthcoming.

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.

Hughes, M. A. and 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.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. 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*, pages 128–136, New Orleans, USA.

Kassarjian, H. H. (1977). Content analysis in consumer research. *Journal of Consumer Research*, 4(1):8–18.

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

Keusch, F. (2014). The influence of answer box format on response behavior on list-style open-ended questions. *Journal of Survey Statistics and Methodology*, 2(3):305–322.

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.

Kurasaki, K. S. (2000). Intercoder reliability for validating conclusions drawn from open-ended interview data. *Field Methods*, 12(3):179–194.

Lee, L. H., Wan, C. H., Rajkumar, R., and Isa, D. (2012). An enhanced support vector machine classification framework by using Euclidean distance function for text document categorization. *Applied Intelligence*, 37(1):80–99.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–50, New York, USA.

Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, volume 33, pages 81–93.

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

Mannetje, A. and 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., 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., and Holcekova, M. (2018). Machine learning and verbatim survey responses: classification of criminal offences in the crime survey for England and Wales. In *Big Data Meets Survey Science Conference*, Barcelona, Spain.

McLauchlan, C. and Schonlau, M. (2016). Are final comments in web survey panels associated with next-wave attrition? *Survey Research Methods*, 10(3):211–224.

Meitinger, K., Braun, M., and 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.

Montgomery, A. C. and Crittenden, K. S. (1977). Improving coding reliability for open-ended questions. *Public Opinion Quarterly*, 41(2):235–243.

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

Oberski, D. (2018). Can Facebook "likes" measure human values? In *Big Data Meets Survey Science Conference*, Barcelona, Spain.

Patel, M. D., Rose, K. M., Owens, C. R., Bang, H., and 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.

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

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

Schierholz, M. (2019). *New Methods for Job and Occupation Classification*. PhD thesis, University of Mannheim. `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. `https://doi.org/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. and Couper, M. P. (2016). Semi-automated categorization of open-ended questions. *Survey Research Methods*, 10(2):143–152.

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

Schuman, H. and Presser, S. (1979). The open and closed questions. *American Sociological Review*, 44:692–712.

Severin, K., Gokhale, S. S., and Konduri, K. C. (2017). Automated quantitative analysis of open-ended survey responses for transportation planning. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, pages 1–7, San Francisco, USA.

Singer, E. and Couper, M. P. (2017). Some methodological uses of responses to open questions and other verbatim comments in quantitative surveys. *Methods, Data, Analyses*, 11(2):115–134.

Smyth, J. D., Dillman, D. A., Christian, L. M., and McBride, M. (2009). Open-ended questions in web surveys: Can increasing the size of answer boxes and providing extra verbal instructions improve response quality? *Public Opinion Quarterly*, 73(2):325–337.

Spooren, W. and Degand, L. (2010). Coding coherence relations: reliability and validity. *Corpus Linguistics and Linguistic Theory*, 6(2):241–266.

Stefanski, L., Wu, Y., and White, K. (2014). Variable selection in nonparametric classification via measurement error model selection likelihoods. *Journal of the American Statistical Association*, 109(506):574–589.

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

Wang, D. J., Shi, X., McFarland, D. A., and Leskovec, J. (2012). Measurement error in network data: a re-classification. *Social Networks*, 34(4):396–409.

Wang, Z., Sun, X., Zhang, D., and Li, X. (2006). An optimal SVM-based text classification algorithm. In *2006 International Conference on Machine Learning and Cybernetics*, pages 1378–1381, Dalian, China.

Weber, R. P. (1990). *Basic Content Analysis*. Number 49. Sage.

Ye, C., Medway, R., and Kelley, C. (2018). Natural language processing for open-ended survey questions. In *Big Data Meets Survey Science Conference*, Barcelona, Spain.

Zade, H., Drouhard, M., Chinh, B., Gan, L., and Aragon, C. (2018). Conceptualizing disagreement in qualitative coding. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, Montreal, Canada.

**

Appendix A

Coding Manual of the Patient Joe Data Set

February 2013

Coding Manual for the Dutch "Patient Joe" Data set

Matthias Schonlau,Ph.D.
Schonlau@uwaterloo.ca
419-888-4567 x31518

Setup:
Survey respondents were given the following hypothetical scenario:

"Joe's doctor told him that he would need to return in two weeks to find out whether or not his condition had improved. But when Joe asked the receptionist for an appointment, he was told that it would be over a month before the next available appointment."

What should Joe do? "

(there are additional questions which we will ignore here).

The answer should be coded in one of four categories:

1. proactive
2. somewhat proactive
3. passive
4. destructive

The text below explains what each category means and gives examples.

General coding rules:

- No response: Do not code anything. When responding only to the original question but not to the probe, do not code anything for the probe
- Multiple answers: If an answer contains two responses (e.g. one passive and one proactive), code the more active one. Do not code multiple responses.
- Misunderstood questions: (1) If the answer reveals that the respondent did not understand the question correctly, try to code the degree of activation evident within the patient's understanding. E.g., if the patient believes a nurse tries to cancel an already existing appointment, having the nurse double check with the doctor is a proactive response. (2) Flag all misunderstood questions in a separate column (this should be quite rare)

Code for Active Role:
Proactive [Problem solving Self Advocate]: *check with doctor*
Patient takes active steps towards getting an appointment in two weeks. Steps have to be taken before leaving the doctor's office. Requesting that staff double check with the doctor while also accepting the appointment after the two week window is acceptable. It does not matter whether

the patient is polite or impolite in accomplishing his/her goal (within reason). Examples of proactive roles are:

- Insisting on getting the appointment in two weeks and if the receptionist won't book it, then ask to speak directly to the doctor.
- Insisting on getting the appointment in two weeks ("insisting" implies that the patient will not give up, it is presumed eventually somebody will check with the doctor)
  - But: "Telling the staff that the doctor said the appointment should be within two weeks" (This is rated "passive", it is presumed that patient will accept the appointment without anybody checking with the doctor).
- Asking staff to double check with the doctor first
- Yelling at the staff until they give an appointment in two weeks
  - But: "Yell at the staff and leave if they do not give you an appointment" (This is rated "destructive" because it allows for a reasonable chance of failure/ getting no appointment.)
- Going back to the doctor
- Asking to be rescheduled with another doctor in the ***same*** practice. (This is reasonable because there is continuity of care within the same practice.)
- Taking the appointment at 4 weeks while also insisting that staff confirm with doctor that 4 weeks. (As long as staff agrees to check with the doctor while the patient is still in the office, checking with the doctor can occur after the patient has left. In this case, they were proactive in engaging doctor in the decision making process. It is reasonable to assume that if doctor said no, they would be given two week appointment or person would not have taken appointment.)
- Wait until two weeks are up and just come back. (While not the ideal proactive approach, this patient is making sure he/she will be seen in two weeks)
- Ask for a referral to another doctor (While not the ideal proactive approach, the patient is taking active steps and the word "referral" implies that he/she is trying to stay inside of the system).

Somewhat active: *call me if*

Patient accepts the appointment but asks to be called, or patient accepts the appointment and asks the staff to check with the doctor later, or patient asks staff to make an exception. Examples are:

- Leaving his/her name and number with the receptionist in case an appointment comes up.
- Asking to be called if there is a cancellation
- Leaving the office with the intention to later call the office to check that the appointment is ok with the doctor.
- Asking to squeeze patient in
- Asking to double book
- Asking for other alternatives

Passive role

Patient takes no action that has a reasonable chance attaining patient's goal. Merely telling the staff what the doctor said (without insisting the staff check with the doctor or taking other more active steps) is considered passive. Examples are:

- Take the next available appointment
- Accept the appointment
- Do not be a troublemaker (or: do not interfere with doctor's office scheduling) and take the appointment
- Tell the staff that the doctor wanted an appointment in two weeks.
    - But: Tell the staff that the doctor wanted an appointment and insist that they double check with the doctor. (This is proactive).
    - But: Insist to get a 2 –week appointment (This is proactive).
- It depends on the severity of the condition (There is a reasonable chance of not getting a 2 week appointment).

Destructive: *go elsewhere*

Patient leaves established care to go to another doctor or patient leaves without any appointment. This category also contains unhelpful actions (e.g. threats) that leave open a reasonable chance of taking no appointment, or leaving to another practice.

- Joe should call somewhere else to get scheduled before that two weeks transpire [go to a different doctor's office]
- Go seek help somewhere else (e.g. urgent care clinic)
- Threaten to leave if patient does not get an appointment

Screen shots from the survey in the LISS panel

Size 1Text box

Additional Coding examples

05-14069-30	He should talk to his doctor and get his two week appointment.	proactive
05-14115-10	First thing I'd do is ask to speak to the doctor. And, unless my condition seemed to be getting worse, I'd probably just wait the month.	proactive
05-14115-30	Joe should tell the secretary that the doctor said she needed to see him in two weeks.	passive
05-14153-20	Ask to see the manager or whatever to make sure that he got to see the doctor within two weeks.	proactive
05-14201-10	Joe should go talk to his doctor.	proactive
05-14201-20	He should tell the receptionist that the doctor gave him an appointment in two weeks, because he won't be able to see him for an additional two weeks, and something could've arised [sic] by then.	passive
05-14215-10	What's Joe's condition? It doesn?t say. He should seek a second opinion, whatever it would be.	destructive
05-14215-20	Explain to the receptionist that the physician wants him back in two weeks, and if that doesn't help, then she needs to put him on a standby list in case there's a sick call or cancellation or something like that.	somewhat
05-14225-10	Tell the receptionist that the doctor recommends he come back in two weeks?two to three weeks.	passive

Appendix B

Coding Schema of the Happiness Data Set

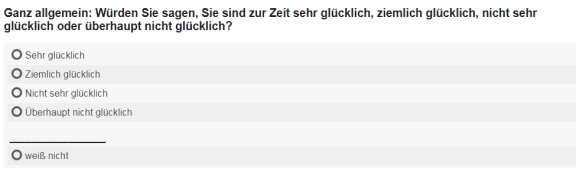
Coding Schema “Aspects: Happiness”

Author: Katharina Meitinger (Version I: 08.01.2018)

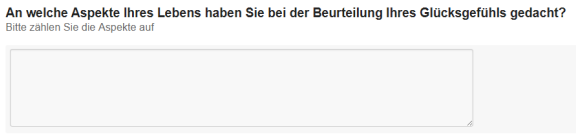
General Information

Coding schema captures the themes that respondents mentioned in the answer box of the specific probe. The specific probe was directly asked after the close item.

Closed Item:



Specific Probe:



Coding instructions

- **Multiple coding:** Coding of more than one answer category per response is possible
- **Multiple coding of same code per response:** Due to the methodological focus of the analysis, it is important that each mentioned topic has to be coded. This means that responses that mention several times the same code, these codes are coded multiple times
- Always use the **most specific category** possible. The general category is only coded, when it is mentioned as such or no suitable specific applies. (Example: “derzeitige allgemeine Situation Beruflich sowie Privat.“ → Coded as „Job“ und „General social network & surrounding“ not coded as „General reference to present”)

- Some categories are exclusive. This means that they cannot be coded in combination with other codes. The following codes are exclusive: 91; 92, 931-940;950

Nr.	Code	Description	Examples:
Social network & surrounding			
1	**General social network & surrounding**	General description of social contacts/relationships: Remark: No specific relationships/people are mentioned	• (kein) Privates/ Privatleben • Privates Umfeld & Situation • Private Sorgen • Soziale Kontakte & Lage •Zwischenmenschliches •Andere Personen •Liebe/ liebe menschen um mich •Persönliche Beziehung •Nachbarschaft
11	**Family**	General mentioning of own family Remark: No specific family members are mentioned	•Familie/familiäre Situation/ •Glücklich mit der Familie •Meine Angehörigen
111	Children	Own children of respondent Remark I: Also the prospect of having children in the future Remark II: Also the absence of children & having no children	•Tochter, Sohn •Baby •Kind/er •Bekomme nachwuchs/Elternzeit •Meine Kinder wohnen nicht mehr bei mir
112	Other relatives	Other relatives that are not covered by code 1-111: Grandchildren, parents, grandparents, siblings, cousins, uncle, aunt , etc. Remark: Death of relatives not coded as code 113 (→ coded as life event: death)	•meine Mutter beansprucht mich •Schwester •Enkel/ Enkelkinder •Oma geworden
12	**Relationship status**		
121	In a relationship	Respondent has: • Partner • Husband/wife/is married • Girlfriend/boyfriend	•Partner/Partnerschaft •Frau/Mann •Freund/in •Familienstand/ Ehe •Beziehung (aber persönliche Beziehung → General social network & surrounding (1))
122	Single	Respondent is single	•Kein Partner/keine Freundin •Da ich mein Leben selber meistere und keinen Partner dafür brauche

			•Single
13	**Further social surrounding**	Respondents mentions friends or pets or other social contacts (not family)	• Freunde Haustier • Hund, Katze
		Health	
2	**General health**	Respondent refers very generally to health (being healthy/being sick) No distinction between physical and mental health	• Gesundheit/ Gesundheitliche Verfassung & Beschwerden • momentaner Gesundheitszustand • Krankheit/ krank/ nicht krank/ Kranheit Angehörige • allgemeines Wohlbefinden/Wohlergehen
21	**Physical health**	Respondents mention health issues regarding their **bodies** References to health condition of others also coded here	• **Specific diseases**: Krebs/ Erkältung/ Allergien • **Chronical conditions**: Chronisch krank • **Pain:** Schmerzen • **Treatment:** Zahnoperation • **Physically disabled:** Schwerbehinderung • **Physical shape:** Gewicht/ Sportlichkeit/ Nicht fit/ Müde/ Aussehen • **Age:** Alter Älterwerden • Sex/ Sexualverhalten
22	**Mental health**	Respondents make references to: - **Mental health** issues - **Mood** (general, positive & negative) - **Loneliness**	**Mental health:** • Depression/ Psychiatrie • mein Leben ist nicht lebenswert/ Hoffnungslosigkeit/ Enttäuschung • Stimmungsschwankung/ Keine Selbstsicherheit/ Gefühl nur Fehler zu machen/ Gefühl das über einen gelästert wird/ Mobbing am Arbeitsplatz **Mood:** • **General:** An den inneren Zustand/Gefühlslage & -ebene, allgemeine Laune, wie es mir im Moment psychisch geht/ mein Befinden/Persönlichkeit • **Positiv:** innerer Friede & Ruhe, Freude, Humor Ausgeglichenheit, kein Stress, positiv und gut gelaunt, friedliches Leben/ weil ich Optimist bin, Zufriedenheit • **Negativ:** Angst, Sorgen, Stress, Viel zu tun/ Finde das es zurzeit immer negativer wird zu leben **Loneliness:** Einsam/ Allein
		Job	
3	**Job situation**	Respondents refer to their job	• Berufssituation/Arbeitsmäßig/Arbeit

		situation	• Mein Chef • Arbeitsleben • Job/Nebenjob/ mieser Job neuer Job ab Dezember • Workload/Ich arbeite 12 Tage durch. Habe 2x im Monat am WE frei...Unter der Woche nie einen freien Tag • Kaum Erfolgserlebnisse/ Fehlschläge
31	**Not employed**	Respondents are not currently employed. This can have different reasons: Unemployment, retirement, still in school or university	• **Unemployed**: Arbeitslosigkeit, Jobcenter • **Retirement**: Pension/Rente/ Teilrente/ ich muß nicht mehr zur Arbeit/ Altersversorgung • **Studen**t: Studium/Uni • **School & apprenticeship:** Stress in der Fortbildung/ Neue Schule/ Warten auf die Ausbildung /Schulstress/ Großer Lernaufwand • **Not able to work**: Arbeitsunfähig
Financial situation			
4	**Financial situation**	Respondents discuss whether they have money or not. This can refer to different dimensions: • General financial situation • Fortune, assets & wealth • Income • Poverty, debts & credits	• **General financial situation:** finanzielle Situation & Lage, Finanzen • **Fortune, assets & wealth:** Vermögen, Geld, Abgesichert, Eigentum, Versorgungssicherheit, Schuldenfrei • **Income:** Einkommen/ Gehalt/ Versorgung/ Schlechte Bezahlung/ Verdienst/ verdiene gutes Geld • **Poverty, debts & credits:** Armut/ Altersarmut/ Insolvenz/ Kredit/ Geldproblem/ finanzielle Sorgen/ Kosten/ Autoreparaturkosten/ nicht genug Geld zum Leben/ das lebein ist nicht grade lustig wenn man nichts hat.
Life situation & living conditions			
5	**General life situation & living conditions**	Respondents refer very generally to life situation or living conditions	• Lebensstandard/ -qualität/ -stil • Lebensweise/ -umstände/ -situation/ -bedingungen
51	**Housing**	Respondents refer to their housing situation, e.g., their flat, their house, having to move etc.	• **General:** Behausung/ Wohnsituation/ Wohnraum • **Flat:** Wohnung, Wohnungsnot • **House:** Eigenheim/ eigenes Haus **Moving:** Umzug

52	**Leisure time & hobbies**	Respondents mentions having (or not) leisure time or specific leisure time activities or hobbies	• **General:** habe mehr/zu wenig Freizeit • **Leisure time activities:** Musik/ Aktivitäten/ Spiele/ Filme/ Kunst/ Literatur/ Satire/ Wrestlingshow • **Hobby:** Hobby/ Fußballverein/ Prüfung in meinem Sport • **Voluntary work:** Ehrenamt/ zu wenig Zeit fürs Ehrenamt
53	**Further aspects living conditions**	Respondents mentions aspects such as holidays, own car, food	• **Holidays:** USA reise, Urlaub • **Car**: Auto • **Food**: Essen & Trinken/ keinen Hunger
		Politics, security & society	
6	**Politics, security & society**	Respondents refers to politics, specific political issues, society or security issues	**Politics:** Politik/ Weltpolitik/ allgemeine politische Lage in Deutschland/ politische Umstände **Specific political issues:** Flüchtlinge **Society**: Gesellschaft/ soziale Ungerechtigkeit **Security**: • Frieden/ Sicherheit • Kriminalität/ Gewallt/ Terror/ zunehmende Gewalt auf den Straßen
		Life event	
7	**Life events**	Respondents mention relevant/important life events: • Marriage • Divorce • Death	**Marriage:** Anstehende Hochzeit **Separation & divorce:** Scheidung, Trennung/ Ex/ von meiner großen Liebe verlassen **Death** • Trauer/Verlust/ Leid/Todesfall • Verstorbene/tote Frau/ Bruder/ Ehemann/Opa • ich habe einen geliebten menschen in meiner familie verloren durch eine schwere krankheit
		Time references	
8	**Reference to time**	Very general remark to life/situation • in the past • in the present • in the future Respondents think about different seasons, specific weekdays or holidays or specific weather	Example reference to past • an mein bisheriges leben • ich war gerade in Erinnerungen Example reference to present: • Momentane/aktuelle/derzeitige Jetziger Zustand/ Situation/ Erlebnisse • Wie es mir im Moment geht • Es läuft zurzeit super/ Zur Zeit ist insgesamt alles ok • dieses Jahr war ein sehr chaotisches Jahr • Zeit

			Example reference to future: • Zukunft/-pläne/ -perspektive • Perspektiven/ keine lebensperspektive Reference to calendar year & weather: • Sommer/dunkle Jahreszeit • Freitag /Sonntag /Wochenende • Wetter/ Kalt/ Sonne • Weihnachten
Rest			
91	**All**	Very general statement regarding all live aspects No specific aspects mentioned Exclusive category	• im allgemeinen/allgemein • Mein Leben Mein gesamtes Leben • Glücklich in allem, • Alles/ weil alles passt/ klappt alles • meine gesamte Situation, • Keine Probleme keine bestimmten Aspekte • allgemeinen Situation
92	**None**	Very general statement regarding no specific live aspects No specific aspects mentioned Exclusive category	• an nichts/nichts/ nichts spezielles • Keine/ / keine besondere/an keine ich fühle mich einfach gut /
93	**Substantive rest category**	Any substantive response that is not covered by codes 1-84	• Tierleid • Ich • Mädchen • Warten • Besser als andere • Reaktivierung im Landesdienst NRW Sinn im Leben/ kann mein Leben selbst gestalten/ Freiheit/ Selbstverwirklichung • Man kann immer etwas verändern • Qualität • kontinuitaet • Pilze Bäume Umwelt • Leben • Lebensvorstellung /Erwartungen

Problems & Nonresponse			
910	**Problem with question**	Respondents critizices the question	• Was versteht ihr unter Glücklich • Ich weiß nicht was sie meinen • ? • das glück nur ne momentaufnahme ist, also doofe frage •
920	**Reduced Motivation**	Respondent expresses a reducing motivation or increasing frustration with the number of probes.	• Darauf habe ich im vorangehenden Abschnitt geantwortet.“ • “Auf diese Frage habe ich schon geantwortet. • “Ich habe sie im vorangehenden Absatz schon beschrieben.“ • “answered this under previous question!”
931	**Complete Nonresponse**	Complete nonresponse: Respondent leaves a blank text box (-99 responses in Excel for all 3/5/10 answer boxes) Exclusive category	• -99
932	**NR: No useful answer**	No useful answer: response is not a word Exclusive category	• Dfgjh/ Kmsdnba/ erdtfzg • 65467978 • ---------- -/- •
933	**Don’t knows**	Don’t know responses Exclusive category	• Keine Ahnung • Kp kein Plan •
934	**Refusals**	Respondents refuses to provide a response Exclusive category	• keine aussage/ ka • nein • Möchte ich nicht näher erklären • NUR normale fragen • geht das jemanden etwas an?
935	**Other nonresponse**	Responses that are insufficient for substantive coding Exclusive category	• Einfach so • Weil halt Ehrlich/ klar/ genau
936	**Repetition of answer categories**	Respondents just repeat the wording of the answer categories Exclusive category	• sehr glücklich, Ziemlich glücklich • Ich bin glücklich, Glücklich/Glück/ Glücklich in allem • Bin voll u. ganz glücklich

940	**Skipped answer boxes**	Respondent does not start with the first answer box but start writing in the 2,3,4,5 answer box Exclusive category	Coding instructions: Please code Code 940 in each empty answer box till the answer box is filled in Example: Respondents only starts in answer box number 3. Please code Code 940 for answer box 1 &2 (not for boxes 4 and following)
950	**Respondents that broke up before submitting any response to open-ended question**	Respondents that: - Received question: Correct number in trigger variable (e.g., 1 for control group, 2 for experimental group 1, 3 for experimental group 2) - Provided response to closed item (dupl1_v_69): 1-7;97 - Quit survey before responding: o -66 values at open-ended variables o 22 value at dispcode variable	

Appendix C

Coding Schema of the Democracy Data Set

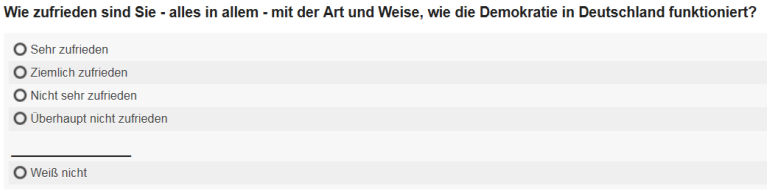
Coding Schema “Democracy”

Author: Katharina Meitinger (Version : 07.06.2018)

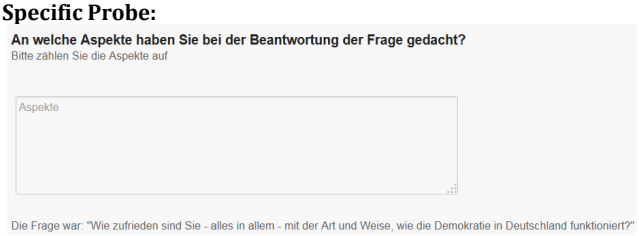
General Information

Coding schema captures the themes that respondents mentioned in the answer box of the specific probe. The specific probe was directly asked after the close item.

Closed Item:



Specific Probe:



Coding instructions

- **Multiple coding:** Coding of more than one answer category per response is possible
- **Multiple coding of same code per response:** Due to the methodological focus of the analysis, it is important that each mentioned topic has to be coded. This means that responses that mention several times the same code, these codes are coded multiple times
- Always use the **most specific category** possible. The general category is only coded, when it is mentioned as such or no suitable specific applies. (Example: “)
- Some categories are exclusive. This means that they cannot be coded in combination with other codes. The following codes are exclusive

Inhaltliche Themen

– Akteure & Gruppen –

Beschreibung Codes: Befragter nennt spezifische Akteure oder gesellschaftliche Gruppen ohne auf zusätzliche Aspekte zu verweisen, die durch andere inhaltliche Codes abgedeckt sind

Code	Akteur & Gruppen	Beispiele
101	Politiker Code wird vergeben, wenn keine zusätzlichen Eigenschaften von Politiker genannt werden (sonst → „Eigenschaften Politiker“)	• Politiker/ Politik • Abgeordnete • Vom Volk gewählte Vertreter
102	Parteien Code wird vergeben, wenn nur spezifische Parteien genannt werden. Kommentare bzgl. Regierungsbildung und Mehrparteiensystem → „Demokratische System“	• AFD • SPD gut zugelegt • CDU • Die Linke • An die Grünen • Die Partei (Satirepartei)
103	Ausländer, Flüchtlinge & Asylanten Code wird vergeben, wenn nur die Gruppe der Asylanten, Flüchtlinge und Ausländer benannt wird. Kommentare bzgl. der Ausländer und Flüchtlingspolitik → „Politikfelder“	• Asyl/ Asylanten Asylbewerber • Flüchtlinge • Einwanderer • Zu viel Ausländer und Migranten/ zu viel Multi Kulti • Für Ausländer wird alles geändert - scheisse • Wirtschaftsflüchtlinge
104	Mehrheitsbevölkerung (=Deutsche)	• Als Deutscher wird man bezüglich Wahrnehmung an öffentlichen Stellen benachteiligt • Zu wenig für das eigene Volk • Urdeutsche werden benachteiligt
105	Weiterer Akteure & Gruppen	• Menschen • Staatsbürger • Elite • Muslime/Islam • Freunde • Behinderte Menschen • Alte Menschen

- Politikfelder -

Beschreibung Codes: Befragter nennt spezifische Politikfelder (=Policies)

Code	Politikfelder	Beispiele
201	Allgemeine polit. Lage	• Politische Lage allgemein/ politisch/ Politik an sich läuft sehr schlecht • tatsächliche Politik/ die Politik muss sich ändern/ Arbeit der Politik/ Ich habe an die vielen negativen Nachrichten aus der Politik gedacht • Politische Skandale • Inneres
202	Außenpolitik	• Führungsrolle in Europa/ EU • G 20 • Türkei/ Russland/ … • Geld an Griechenland • Vergleich mit anderen Ländern
203	Ausländer & Flüchtlingspolitik Code wird vergeben, wenn Themen der Flüchtlings-/Ausländerpolitik angesprochen werden. Bei alleiniger Nennung von Flüchtlingen und Ausländern → „Akteure & Gruppen“	• Ausländerpolitik/ Ausländerzuzug • Flüchtlingspolitik/ Flüchtlingskriese/ Flüchtlingsproblem • Bearbeitung von Asylbewerber • Einwanderung/ zu viel Multi Kulti • Minderheitenschutz • Asylbetrug
204	Finanzpolitik Code wird bei Hinweisen auf Finanzpolitik, Steuern und Staatsverschuldung vergeben.	• Steuerpolitik/ Finanzpolitik • Steuern/ zu hohe Steuern • Steuerverschwendung/ Steuergelder werden unsozial verschleudert • Gelder/Geld/ Geldtreiberei • Solidaritätsabgabe/-zulage • Defizite/ Staatsverschuldung • Bereicherung • Fördermittel • Steuerflüchtlinge
205	Rentenpolitik Code wird bei Hinweisen auf Rentenpolitik und Rente vergeben	• Rente/ Rentensituation • Rente mit 63 • Plünderung der Rentenkassen • Meine Mieterin die 690,-€ Rente hat und keine Grundsicherung beantragt, weil ihrem Sohn der Unterhalt aufgebürdet werden soll
206	Umweltpolitik	• Umwelt/ Umweltschutz • Tierschutz • Atomausstieg • Energiepolitik • Klimawandel
207	Gesundheitspolitik/ -system	• Gesundheit • schlechtes Gesundheitswesen/ Gesundheitsreformen • Krankenversicherung
208	Familienpolitik	• Familienpolitik/ Familien benachteiligt/ Der Staat sollte viel mehr für Familien sorgen • Kinder/ KITA • Alleinerziehende Mütter
209	Weitere Politikfelder	• Rüstungsexporte/ Waffenexporte • Medienpolitik/ GEZ/ Zwangsgebühr • Cannabislegalisierung

- Situation -

Beschreibung Codes: Befragter nennt Aspekte der aktuellen gesellschaftlichen Lage in Deutschland

Code	Bereiche	Beispiele
301	Allgemeine Lebensbedingungen / Lebensstandard	• Lebensumstände/ Lebensbedingungen/ Lebensart/ -qualität/Zurechtkommen in Deutschland • Gesellschaft • Wohlbefinden/ Wie es mir/einen geht/ Mein Leben/ Meine Situation/ Harmonie im Leben • Zu viele Probleme/ Probleme des Landes/ Probleme der Bürger
302	(Un)gleichheit	• Es gibt (Un)gleichheit/ Ungleichheit der Menschen • ungleiche Behandlung/ es werden nicht alle gleichbehandelt • Un/gleiche Chancen & Risiken • zu große Unterschiede/ gesellschaftliche unterschiede • Gleichberechtigung/ Gleichbehandlung/ Gleichstellung • Frauen haben (nicht) die selben Rechte wie Männer • Ältere Menschen werden schlecht behandelt, haben teilweise ewig gearbeitet
303	(Un)Gerechtigkeit	• Ungerechtigkeit/ Soziale (Un-) Gerechtigkeit/ Gerechtigkeitsdiskussionen • ungerechte Einkommensverhältnisse • unfair • gewisse Gruppen werden bevorzugt
304	Wirtschaft	• Wirtschaft/ Wirtschaftlich/ Wirtschaftliche Situation • Wirtschaftswachstum/ die Wirtschaft floriert • Wohlstand • Finanzielle Absicherung/ finanziell • Kapitalismus wird als einzige Wirtschaftsmöglichkeit angesehen/ Kapitalismus • die wirtschaftlichen Aspekte sind zu wichtig/ Schutz der Wirtschaft, nicht der Menschen/ Wirtschaftsinteressen haben leider Priorität vor Sozialem oder Umweltfragen
305	Arbeitsmarkt	• Arbeit/ Arbeitslosigkeit/ Es gibt immer weniger Arbeit • zu wenig Lohn/ Lohnunterschiede/ Gehalt/ Einkommen • Arbeitnehmerfreundlichkeit • Tarife/ Mindestlohn • Mein Berufsstand/ Hartz4/ Selbstständigkeit • Einkommen bei Frauen und Müttern • Gewerkschaften
306	Armut / Reichtum	• Es gibt viele Arme/ Zuviel Armut/ Arm und Reich Spanne/ Reichtum • Kinderarmut/ Kinder in Not/ Altersarmut/ An die Rentnerin die bei Aldi nicht viel kaufen kann • Es gibt Obdachlose • Menschen in Not • Gute Lebensmittel/ Ich muss nicht hungern

307	Integration, gesellschaftlicher Zusammenhalt & Austausch untereinander	• Integration/ Akzeptanz/ Toleranz/ Respekt • Ausländerfeindlichkeit • Ausgrenzung • Umgang mit anderen Kulturen/ Kultur/ Miteinander • Flüchtlinge passen sich nicht an • Hilfe wo Hilfe gebraucht wird/ Solidarität/ gesellschaftlicher Zusammenhalt/ Menschlichkeit • es gibt viele positive Diskussionen/ Streitkultur
308	(Un)Sicherheit & (Un)Ordnung	• Sicherheit • Frieden, gesellschaftlicher Frieden/ kein Kriegsgebiet/ Leben ohne Angst • Zu viele Gefahren/ existierende Kriminalität/ Terrorbekämpfung/ Gewalt • Ordnung/ Chaos • Stabilisierung von Deutschland • Überwachung
309	Stagnation & Zukunft	• Abwartehaltung in allen Dingen • Es ändert sich dadurch nicht wirklich was/ dass es bald mal losgehen soll • Wir können nichts ändern • es bleibt immer wie es ist/ egal was man wählt... nichts wird sich ändern • es wird immer nur geredet • keine andere Wahl • Fehlende Innovation • Zukunft
310	Weitere Bereiche	• weil die Mittelschicht überhaupt nicht mehr wahrgenommen wird
311	Bildung	• Bildung/ Bildungschancen/ Bildungsunterschiede/ Bildungssystem • (zu wenig) Politische Bildung • Schule/ Ausbildung
312	Verkehr & Infrastruktur	• Transport/Verkehr/ÖPVN • (verrottete) Infrastruktur
313	Wohnungs-/ Mietsituation	• Wohnungsmarkt/ Wohnungsnot/ Wohnsituation • (Zu hohe) Mieten/ Mietpreise/ Mietwucher • Mietpreisbremse • (sozialer) Wohnungsbau/ bezahlbarer Wohnraum

- Beurteilung Verhalten Politiker & Parteien -

Beschreibung Codes: Befragter bewertet/nennt unterschiedliche Verhaltensaspekte von Politiker

Code	Eigenschaften	Beispiele
400	(Un)qualifizierte Politiker & Kompetenzen	• unfähige Politiker/ Uneinsichtigkeit der Politik • Kluge, sympathische Art von Fr Wagenknecht • Wieso braucht man überall eine Ausbildung nur Politiker brauchen keine !!!!! • Politiker sind oft unqualifiziert/ Qualifikation der Politiker/ Seriosität der Politiker/ Zu wenig Fachkenntnis bei Politikern
401	Eigeninteresse & Gier vs. Gemeinwohl	• Weil jeder auf seinen eigenen Vorteil bedacht ist/ Politiker sind auf ihren eigenen Vorteil bedacht • Egoistisch/ Profitgier/ Machthungrig • Posten Geschacher • Missbrauch der Demokratie durch Politiker • Alle wollen absahnen und niemand hat den Mut das notwendige zu tun/ Sie bestehlen uns von allen Seiten • nichts Passiert zum Wohle des Volkes/ Ausrichtung der Politik am Gemeinwohl
402	Interesse an Volk/Bürger & Repräsentation	• Politiker/Parteien sind nicht volksnah/ Kommunikation der Politiker/Parteien • Die Parteien vertreten zu selten die Menschen, fokussieren sich auf die Wirtschaft/ was die Bürger sagen, interessiert den Parteien nicht. • das Volk wird nicht gefragt/ die Bürger werden übergangen • Politiker/Parteien interessieren sich nicht für einzelne Menschen • Politik, die wichtige Aspekte ignorieren/ Politik kümmert sich zu wenig um die Armen und Kranken • Vertretung/ das Volk wird nicht richtig vertreten
403	(keine) Einhaltung von Wahlversprechen	• Wahlversprechen/ Versprechen/ Politiker versprechen zu viel • Nach der Wahl achten sie nicht den Wählerwillen • Politiker halten sich nicht an ihr Wort
404	kurz/langfristige Zielsetzung	• Die aktuelle Politik hechelt jedem Trend nach und wechselt ständig die Meinung • kein Weitblick in der Politik • Keine klaren Konzepte
405	(kein) Lobbyismus	• Lobby • Politische Klüngelei mit Wirtschaft/ Macht nur bei Reichen und Konzernen / Industrie zu mächtig
406	(keine) Korruption	• zu viel Korruption/ korrupt • Bestechlichkeit
407	(Un)ehrlichkeit	• dass Politiker nicht ehrlich sind/nach der Wahl wird gelogen • unglaubwürdig/ Glaubwürdigkeit der Parteien • Ehrlichkeit/Anstand • Verblendung durch Politiker • man fühlt sich verarscht • üble Nachrede in der Politik
408	Transparenz	• Transparenz • Medien verdrehen einiges/ Medien versuchen Meinungsbildung zu beeinflussen statt zu berichten • Informationsfluss
409	Allgemein/ Weitere Eigenschaften Politiker	• Verhalten der Politiker • Die materielle Sicherheit für Abgeordnete ist total übertrieben/ Nebenverdienste der Volksvertreter/ Bezüge der Politiker (extra Kategorie)

- Demokratische System -

Beschreibung Codes: Befragter bewertet/nennt unterschiedliche Aspekte des demokratischen Systems

Code	Eigenschaften	Beispiele
500	Allg. Statements bzgl. System und Funktionsweise	• System/ Demokratie • Deutschland geht es sehr gut - also muss das System ja gut funktionieren • funktioniert nicht so gut/ vieles hat nur den Anschein von Demokratie • mehr Diktatur als Demokratie • Es gibt nichts besseres
511	Wahlen	• durch Wahlen/ Dass man wählen kann/ Auch Dumme dürfen wählen • jede Stimme zählt/ gemeinschaftliche Abstimmungen/ Stimme des Volkes/ Wahlbeteiligung • Bundestagswahl/ 5%-Hürde • Wahlen manipuliert/ gekauft • Hetze gegen Menschen die die AfD wählen • Wahlfreiheit/ ich kann wählen wen ich möchte
512	Mehrparteienprinzip	• Parteien • Das Parteiensystem/ Parteienspektrum/ Parteienvielfalt • es gibt etliche unterschiedliche Parteien/ zu viele Parteien/ zu viele Parteien kommen auf keinen Nenner • Parteien mit knapp 10Prozent haben nichts in der Regierung zu suchen • alle größeren Parteien sind sehr ähnlich • politischer WettbewerbBis jetzt keine zu extremen Parteien/ Stimmen für linken und rechten Rand nicht allzu hoch • nicht-konstruktive Oppositionsparteien • Begrenzung Extremismus/ Keine radikalen Ausreißer links/rechts/ Stimmen für linken und rechten Rand nicht allzu hoch
513	Direkte Demokratie & Partizipation	• Bürgerbeteiligung/ Bürgerentscheid/ Bürgerbegehren/ Bürgerwille • Zur perfekten Demokratie fehlt die Volksabstimmung • zu wenig Mitspracherecht/ zu wenig Mitbestimmungsrecht/ Mitsprachemöglichkeit • Die Bürger sollten mit einbezogen werden • Demokratie analog Schweiz
530	Rechtstaatsprinzip & Gewaltenteilung	• Recht/ Bürgerrechte/ Rechtsstaatlichkeit/ Rechtsstaat/ Rechtssystem/ Gerichtsbarkeit • Gewaltenteilung/ Trennung der Gewalten • Funktionsweise der drei-Gewalten-Teilung
531	Judikative	• ungerechte Richtbarkeit/ Gefühlte Ungerechtigkeit bei Straftaten • Umsetzung von gerechteren Strafmaßen • Bestrafung von Straffälligen • Justiz
532	Exekutive	• Regierung / Regierungsbildung/ Jamaika • Merkel/ Kanzlerin • Die Koaliation ist sie immer noch nicht einig • Polizei/ Polizeigewalt • Umsetzung/ Entscheidungen

		• Verwaltung/ Behörden/ Bürokratie
533	Legislative	• Bundestag/ Landtag/ Bundesrat/ Parlament • Gesetze/ Gesetzgebung teilweise unverständlich Parlamentarismus
540	Grundgesetz & Menschenrechte	• Grundrechte/ Grundgesetz/ Die Verfassung • Menschenrechte
541	Meinungsfreiheit:	• frei Meinungsäußerung/ Meinungsfreiheit/ Redefreiheit • Pressefreiheit
542	Restliche Freiheiten	• frei/ Freiheit/ Dass man sein kann wie man ist • Religionsfreiheit/ Glaube • zu wenig Eigenbestimmung/ Fremdbestimmung/ (staatliche) Bevormundung
550	Sozialsystem	• Unser Sozialsystem/ Sozialstaatsabbau/ Sozialleistungen • Soziales • soziale Absicherung/ soziale Sicherheit/ Soziale Hilfe vom Staat
560	Weitere Aspekte System	• die Menschenrechte werden eingehaltenParteienfinanzierung • Föderalismus/ unterschiedliche Kompetenzen für Länder und Bund in durchaus zentralen, bundesweiten Fragen • Populismus • Fraktionszwang • Beamtentum abschaffen/ Beamtenprivilegien

		Rest	
910	**All** Exclusive category	Very general statement regarding all live aspects No specific aspects mentioned	• im Allgemeinen/allgemein • Alles • Keine Probleme keine bestimmten Aspekte • allgemeine Situation • alles in allen
920	**None** Exclusive category	Very general statement regarding no specific live aspects No specific aspects mentioned	• an nichts/nichts/ nichts Spezielles • Keine/ / keine besondere/an keine
930	**Substantive rest category**	Any substantive response that is not covered by codes	• Psyche • Bereicherung • Ein Land • Macht (Ohne Zusammenhang) • (Deutsche) Vergangenheit • Stolz ohne Nationalstolz
		Problems & Nonresponse	
940	**Problem with question**	Respondents criticizes the question	• Was versteht ihr unter Glücklich • Ich weiß nicht was sie meinen • ? • das glück nur 'ne Momentaufnahme ist, also doofe frage
950	**Reduced Motivation**	Respondent expresses a reducing motivation or increasing frustration with the number of probes.	• Darauf habe ich im vorangehenden Abschnitt geantwortet." • "Auf diese Frage habe ich schon geantwortet. • "Ich habe sie im vorangehenden Absatz schon beschrieben." • "answered this under previous question!"
961	**Complete Nonresponse** Exclusive category	Complete nonresponse: Respondent leaves a blank text box (-99 responses in Excel for all 3/5/10 answer boxes)	• -99
962	**NR: No useful answer** Exclusive category	No useful answer: response is not a word	• Dfgjh/ Kmsdnba/ erdtfzg • 65467978 • ---------- -/- • Ausllä
963	**Don't knows "doesn't can"** Exclusive category	Don't know responses	• Keine Ahnung • Kp kein Plan • Kann ich nicht beschreiben • kann ich nicht genau sagen • Weiß nicht
964	**Refusals**	Respondents refuses to provide a	• keine aussage/ ka

	"doesn't want" Exclusive category	response	• nein • Möchte ich nicht näher erklären • NUR normale fragen • geht das jemanden etwas an?
965	**Other nonresponse** Exclusive category	Responses that are insufficient for substantive coding	• Einfach so • Weil halt • Ehrlich/ klar/ genau • Bauchsache • Bla • gut
970	**Repetition of answer categories** Exclusive category	Respondents just repeat the wording of the answer categories	• Demokratie • Art und Weise/ funktionieren/ Demokratie • Bin /nicht/zufrieden • Zufrieden(heit)
980	**Skipped answer boxes** Exclusive category	Respondent does not start with the first answer box but start writing in the 2,3,4,5 answer box	Coding instructions: Please code Code 940 in each empty answer box till the answer box is filled in Example: Respondents only starts in answer box number 3. Please code Code 940 for answer box 1 &2 (not for boxes 4 and following)
990	**Respondents that broke up before submitting any response to open-ended question**	Respondents that: - Received question: Correct number in trigger variable (e.g., 1 for control group, 2 for experimental group 1, 3 for experimental group 2) - Provided response to closed item (dupl1_v_69): 1-7;97 - Quit survey before responding: o -66 values at open-ended variables o 22 value at dispcode variable	
991	**Answer spreads out over more than one box**	Answers refer to answer in former answerbox	Answerbox 1: I Answerbox 2: Don't Answerbox 3: Know Answerbox 1: Merkel macht einen guten Job Answerbox 2: Sie leistet eine gute Arbeit