How We Code

David Williamson Shaffer(✉) and A. R. Ruis

University of Wisconsin-Madison, Madison, WI 53706, USA
dws@education.wisc.edu

Abstract. Coding data—defining concepts and identifying where they occur in data—is a critical aspect of qualitative data analysis, and especially so in quantitative ethnography. Coding is a central process for creating meaning from data, and while much has been written about coding methods and theory, relatively little has been written about what constitutes best practices for *fair* and *valid* coding, what justifies those practices, and how to implement them. In this paper, our goal is not to address these issues comprehensively, but to provide guidelines for good coding practice and to highlight some of the issues and key questions that quantitative ethnographers and other researchers should consider when coding data.

Keywords: Coding · Automated classifiers · Inter-Rater Reliability (IRR) · Fairness · Validity

1 Introduction

Although it goes by many names—*classifying*, *annotating*, *categorizing, rating, indexing, linking, labeling, tagging, marking up*—the process of coding data is a critical part of data analysis, and particularly of data analysis in *quantitative ethnography*. In brief, *coding* refers to the process of (a) defining concepts of interest ("Codes") and (b) identifying places in a dataset where they occur.

Despite the central role coding plays in interpreting data and testing claims—and the extensive literature on coding methods (e.g., [19]) and theory (e.g., [23])—there has been relatively little written about what constitutes best practices for fair and valid coding, what justifies those practices, and, equally important, how one should implement them.[1] The result is a set of coding practices that range from merely inconsistent to wholly inappropriate, leading to mistakes and misunderstandings in non-standard situations.

There are many reasons for this state of affairs. Different fields have different goals and assumptions about coding, and thus sometimes use different practices. Existing measures for the reliability of coding are problematic, and for historical reasons are used in ways that are not statistically sound. And the theoretical underpinnings of the

[1] Because we conceive of this contribution as an overview of key issues in coding theory and practice for QE researchers, both novice and advanced, and due to the limitations of space, we do not include a comprehensive review of the literature on coding and qualitative discourse analysis. We will address this shortcoming in a future, expanded version of this paper.

A. R. Ruis and S. B. Lee (Eds.): ICQE 2021, CCIS 1312, pp. 62–77, 2021.
https://doi.org/10.1007/978-3-030-67788-6_5

processes of coding are actually quite complex, covering everything from hermeneutics and linguistics to statistics and computer science.

In this paper, we are not attempting to resolve all of these issues, nor to provide a complete accounting of all of the conceptual machinery that explains how and why we code. Rather, our hope is to provide some guidelines for good coding practice, and suggest some of the questions that quantitative ethnographers and other researchers need to consider when coding data.

2 Background

In simplest terms, a Code is a label that we assign to some piece of data. For example, coding a photograph means asserting that it has some particular property of interest (perhaps that it contains the IMAGE OF A PERSON). However, in making what seems like a simple assertion such as this, a researcher is actually engaging in a far more complex process. In particular, Codes have five key properties that need to be accounted for in any sound coding process.

1. **Codes are contextual**. For example, it is easier to identify whether a photograph contains the IMAGE OF A PERSON than it is to determine whether that person is KATSUNGNGAITTUQ. Katsungngaittuq is an Utkuhikhalingmiut word that means something like "pleading" or "whining"—but not exactly. A child whining to have sugar in her tea when there is no sugar in the house is being KATSUNGNGAITTUQ, but so too is a friend who asks you for a favor that makes the you uncomfortable, even if no whining or pleading is involved.[2] Thus, coding is situated within some cultural context: we cannot code without first understanding that context.
2. **Codes are theoretical**. When a researcher codes data, the Codes are no longer situated only in the cultural context from which the data arose. They are also situated in the context within which the research is taking place. This is the well-known distinction between *emic* and *etic* understanding, where the former refers to categories that are meaningful in the community from which the data came, and the latter refers to categories that are meaningful within some theoretical framework. KATSUNGNGAITTUQ is an Utkuhikhalingmiut word, but it was used by an anthropologist to create a theory of *emotion concepts* as a way of understanding how children learn the emotional context of life in Utkuhikhalingmiut culture [1].
3. **Codes are contestable**. Like any claims, claims about whether something contains the IMAGE OF A PERSON or that a person is KATSUNGNGAITTUQ are also inherently *contestable*. That is, we could argue about whether or not any one picture does or does not contain a person, or whether (hypothetically, of course) David is being KATSUNGNGAITTUQ when he asks Andrew to "pick up some sushi for me while you are getting lunch." Moreover, because Codes are contestable, a researcher must be able to provide specific *evidence* to support the claim that a Code applies to some piece of data. The gesture of *pointing* to some specific feature in some specific piece of the data *is* the process of coding, and in simplest terms, every claim about a Code implies some act of pointing, and every act of pointing is a claim about some Code.

[2] This example is drawn from the work of Briggs [1], but see also Shaffer [20].

4. **Data is constructed**. The pieces of data that we code do not represent natural categories or distinctions. They represent a selection of events in the world that were both *recorded* and *transformed* into individual items to be coded. Identifying *segments of data* to be coded also involves some choice. Even the choice to take something as simple as "turns of talk" as individual segments of data is quickly revealed to be a choice and not a natural parsing of the data if one takes a conversation analytic perspective (see, e.g., [12]). This process of breaking data into such pieces, or *segmentation*, raises its own set of questions about how and why it should be done. While the choices that go into segmentation clearly have implications for any coding process, we choose to bracket those questions here and take it as given that data to be coded has been segmented in some defensible way.
5. **Data is finite**. It is only ever possible to code, at any one time, some finite number of specific segments of data. This means that our Codes are not only situated in the culture from which the data came and within some theoretical framework, they are also *anchored* within the data itself. When we code for KATSUNGNGAITTUQ, we are coding for some *cultural phenomenon* (a particular kind of awkward social interaction) that plays a role in some *theoretical framework* (a theory of emotion concepts). But those meanings are also manifest in some particular way or ways *in the specific set of data* that we have. When we code, we are trying to make claims about a Code *as it appears in the data at hand*.

We use Codes because in order to understand something, we have to know what they *mean*. Meaning is fundamental to any human endeavor, but it is particularly salient in the analysis of human behavior. The problem is that the meaning of any single data point is hard to ascribe. And if we are not sure that the underlying data mean what we think they mean, then the patterns that we find do not have much practical use.

In his theory of Discourse [8], Gee makes the distinction between what he calls *small-d [d]iscourse* and *Big-D [D]iscourse*. [d]iscourse is the things that some person or people actually said and did in the world at some point in time—that is, the things we want to understand. [D]iscourse, on the other hand, is the way that some *group* of people—some community—behave in the world: the *kinds of* things they say, actions they take, tools they use, clothes they wear, and so on. We thus have two parallel processes: a process by which data is constructed from [d]iscourse produced by some community of people, and a process by which a theoretical understanding of that community is construed in terms of its [D]iscourse. One exists in the realm of observing and recording events, and the other is a process of interpreting events situated in theory and praxis.

Coding is the key step in the unraveling (and subsequent re-raveling) of meaning because it is the process of bridging these two worlds: the *world of events* and the *world of interpretation*. We refer to the *constructs that make up a [D]iscourse*—the kinds of things that some group of people say and do, to which we attend as researchers and to which they attend as members of some community—as [C]odes: the cultural meanings that some group of people construe, and that we as researchers, in turn, call out to make our theoretical interpretations of that culture. [C]odes are the parts of a [D]iscourse that we want to understand. [c]odes, then, are the *features of the data* we use to identify these constructs: the specific pieces of *evidence* and rules that we use to warrant our [C]odes.

A [c]ode is the collection of features (rules, examples, properties) in the data that we use to claim that the [C]ode is present.

It follows that all coding is an application of *power*. When we attach a [C]ode to some segment of data, we are saying to the data and to the people about whom it was recorded that we are the arbiters of its meaning—at least for the purposes of our analysis. Similarly, we are saying to some theory that we determine whether, when, and how its constructs apply in some setting (and, indeed, which constructs are worth applying).

Our only defense for such an act is that we believe we are doing so *fairly*. That is, the representations we make are such that members of a community we are discussing, proponents of a particular theory or theories, and informed perusers of the data would all feel that our [C]odes are not inaccurate or misleading–that is, we claim what Charmaz [2] refers to as *resonance*.

In this we are also following the work of Goodman [10] (re-elaborated in Shaffer [20]) that all data analysis aims to present some *fair sample* of the world. The concept of a fair sample is based on the idea that all analytical claims are, in fact, re-presentations of some events in the world, a recasting of some original experiences in some digestible form. Such a re-presentation is *fair* if others would feel that (a) it resembles the key features of events in the world, and (b) the things it resembles are not isolated instances, but are similar to some larger pool of events and experiences.

The question then, is how we can (a) choose a set of [C]odes and (b) operationalize them into a set of [c]odes that are fair in this sense. In other words…

3 How Should We Code?

We address this question by starting at the end of the coding process, because beginning with the final product of coding clarifies some of the criteria that guide earlier steps.

3.1 Coded Data

Fairness. The end result of coding is a set of coded data: data with fair claims about some set of Codes for each of its segments. In some cases, this will be data that has been coded by a human rater or raters. In other cases, the data is coded by some automated process.

In human-coded (sometimes called *hand-coded*) data, there are three possibilities: (1) a single rater codes the data alone; (2) two or more raters code some of the data jointly and then one or more of them code some or all of the rest of the data alone; (3) two or more raters code all of the data. In each of these cases, we have to ask the question: What evidence do we have that the Codes are fair?[3] There are two kinds of evidence that are usually marshalled.

The first is that for each [C]ode we provide a *definition* and *examples*. The definition, of course, *is* the [C]ode. The name of the [C]ode is, ideally, descriptive and evocative, but the definition determines the meaning of the [C]ode. That meaning is always enacted

[3] We do not ask: Are the Codes fair? There is no absolute or objective sense in which Codes can be fair or not. But we can ask what evidence we have to support a claim that they are.

within the confines of the data at hand, so the examples provide segments of data to which the [C]ode applies. Expressed in the language of psychometrics, the definition provides evidence of *construct validity* (does the [C]ode relate to underlying theory?) and the examples provide evidence of *face validity* (does the [C]ode make assertions we expect it to make about segments of data?).

The second kind of evidence that [C]odes are fair (in the case of multiple raters) is some measure of *inter-rater reliability* (IRR). There are many measures of IRR that researchers use, and the pros and cons of different measures have been discussed elsewhere [6, 14, 15, 20]. Here we will use Cohen's kappa (κ), as it is common in quantitative ethnography, but whenever we refer to κ we are making a claim about any IRR measure a researcher might choose.

Technically, κ measures the *level of agreement* between two raters taking into account features of the data, or in the language of psychometrics, *concurrent validity* (do two measures of the same construct produce the same results?). But it is easier to think of κ as measuring the *distance* between two raters' understanding of the data. A *high* value of κ indicates that the raters' understandings are close to one another; a low value of κ indicates they are far apart. The distance is proportional to 1/κ.

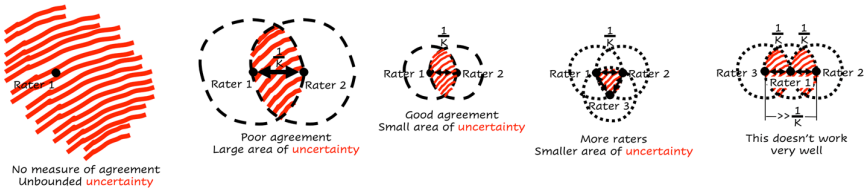

Fig. 1. Low values of κ reflect larger areas of uncertainty about the meaning of a [C]ode than high values of κ; there are diminishing returns for adding additional coders; if there are more than two coders, pairwise measures of agreement are needed.

We can think of this visually as if each rater had a circle around their understanding of the [C]ode, where the radius of the circle is 1/κ (see Fig. 1). The understandings overlap in the area that is less than 1/κ away from both raters. Thus, although we never know the "correct" understanding of a [C]ode (since there is no one correct understanding), κ gives us a way of *bounding the uncertainty* about the meaning of a [C]ode: It is likely to be somewhere in the area of uncertainty between the raters. Thus, a high value of κ gives us confidence that the uncertainty in the [C]ode is small.

This visual representation of how an IRR measure like κ bounds uncertainty also explains why there are diminishing returns to using additional coders. The area of uncertainty will surely be smaller at the intersection of three circles then at the intersection of two—and smaller still at the intersection of 4 or 5 or 6 hyperspheres. But while the area of uncertainty becomes smaller as we add more raters, it still exists. When we go from one rater to two, we go from unbounded uncertainty (that is, we have no measure) to uncertainty that we can quantify and thus choose to accept or reject. Figure 1 also shows why, if there are more than two raters, it is essential to get good agreement between every pair. Otherwise, the uncertainty becomes less bounded rather than more.

Code Validation. Determining fairness is often referred to as a *validating* a [C]ode, although we want to emphasize that fairness implies multiple forms of validation, and how we assess fairness varies depending on how the data has been coded.

All Data Coded by a Single Human Rater. This is most often seen in qualitative studies, and the logic is that any qualitative description of events is inherently perspectival, so adding a second rater (or more raters) does not make the account any more "objective."

In the case of a single rater, the only forms of evidence available are the description of a [C]ode and the examples given. In such a case:

1. Fairness of a [C]ode to existing theory can be warranted by comparing the definition to explanations of the theoretical construct(s) to which it relates.
2. Fairness of a [C]ode to the community that produced the data can be warranted by a positionality statement of the rater, and any other evidence that shows the rater understands the emic perspective of the community.
3. Fairness of a [C]ode to the data can be warranted by the examples given, and by the fact that a single rater has coded all of the data.

All Data Coded by Multiple Human Raters. There are two different ways in which this can be done. Using *social moderation*, each rater codes all of the data and then disagreements are discussed until the raters come to consensus (i.e., $\kappa = 1$) [11, 20]. If social moderation is not used, then κ is computed between the raters.

When multiple raters code all of the data, the warrants for fairness that apply to a single rater can (and should) be used. But we also have information about IRR:

1. Fairness to theory is warranted if the raters come from the same field or theoretical framework. If two raters who understand a theory agree on a [C]ode with (say) $\kappa = 0.90$, then although we do not know that any one segment of data is coded in complete accord with the theory, there is a bound on how far from the theory the [C]ode might be.
2. Fairness to the community is warranted if one (or more) of the raters comes from the community that produced the data. There are circumstances where it might not be possible to have the data coded by a member of the community, and so often community triangulation is done with the analysis as a whole rather than each individual [C]ode. But when possible, including members of the community in the coding process is a strong warrant for fairness to the community.
3. Fairness to the data is warranted by the fact that all of the coders have coded all of the data. κ is an accurate measure of the raters' agreement *for this set of data.*

Multiple Human Raters Code Some Data Jointly and Some Alone. For data coded jointly, all of the warrants apply as when multiple raters code all of the data (and thus also the warrants that apply to a single coder). However, the circumstances become more complex for the data that is only coded by a single rater.

For the *data that is coded by both raters* (DBR), we can measure the rate of agreement *for that data.* If $\kappa(\text{DBR}) = 0.91$, then we know that the rate of agreement for the DBR was 0.91. Notice, however, that the DBR is only a subset of all of the coded data. We cannot

simply assume that because κ(DBR) = 0.91 that κ for all of the data—if we measured it—would be 0.91. That would be as statistically unsound as measuring the heights of 100 students at the University of Wisconsin, taking the average (which when we did it was 1.75 m), and then concluding that the average height of everyone in Wisconsin is 1.75 m.

We have taken a sample from some population (all people in Wisconsin or all the coded data), computed a statistic (either average height or κ), and then assumed that the value of the statistic for the population is the same as for the sample. Instead, we need to control for Type I error (false positives). With height data we might do this using a *t*-test on the sample. But IRR statistics in general, and κ in particular, have distributions whose shapes vary based on the characteristics of a sample; thus it is impossible test them analytically.

We recommend controlling for Type I error using *Shaffer's ρ*, which is an empirical rejective method designed to warrant claims about the κ for a population based on the κ for a sample [6, 7, 20]. The ρ statistic functions much like a *p*-value in a *t*-test: if it is below some critical value (typically $\alpha = 0.05$ or $\alpha = 0.01$), then we conclude that κ for all of the coded data is above some threshold.[4] That is, we use κ and ρ to warrant that the coding done by single raters is fair to the data in the sense that bounding uncertainty now takes into account the fact that our measure was only on a sample of the data.

If we have such a warrant for the rate of agreement over all the data, then we can treat the data coded by a single rater as if it were coded by both raters with a rate of agreement at or above the threshold we specified, with all of the warrants that entails.

Automated Coding. If one of the raters is some automated process—more commonly referred to as an *automated classifier*—then the above still applies, with one exception.

Fairness to theory and fairness to the community are *impossible to achieve with only one human rater and an automated classifier*. The automated classifier does not *understand* the theory and is not a member of the community. Thus, two human raters and one automated classifier are required to use IRR to warrant fairness. The human raters provide warrants to fairness; then acceptable κ and ρ between the automated classifier and each human warrant that the classifier can code the remainder of the data fairly.

Once an automated coding process is determined to be fair, it might be used to code more data that is similar to the data on which it was developed. Technically, this requires a claim of exchangeability: namely, that the original data and the new data are not different in any way that impacts the classifier. We can warrant such a claim by (1) explaining why we think the conditions are the same, or (2) showing that the classifier performs fairly on the new data. Using the first approach means that the fairness of the new coding depends on the accuracy of the assertion of exchangeability. The second requires repeating the coding process: two humans need to code the new data and test agreement with each other and with the classifier.

[4] The ρ statistic can be applied to any measure of IRR with raters using binary codes. There are other statistics that can be used for raters when using non-binary codes.

3.2 Coding Practices

Definition and Examples. The definition of a [C]ode is the claim that is being made about the meaning of a segment of data. We sometimes see brief and relatively uninformative definitions (for example, defining a [C]ode for FRIEND as "any reference to a friend or friendship"). But brief definitions make weak or vague claims. We recommend more robust definitions, such as:

> FRIEND: *Explicit reference to a being a friend or any related terms (e.g., buddy, pal, chum, bestie), or to the general concept of friendship; could include familial terms in relation to a non-family-member, for example, calling someone "my sister" when they are not actually a sister; referring to a platonic relationship of mutual respect, trust, loyalty, and/or honesty using a single term or phrase.*

The definition of a [C]ode should always be accompanied by examples from the dataset. Examples show instances where the definition of the [C]ode clearly applies in a case or cases that are not trivial or obvious. So, in that sense, a segment of data where someone says "You're a good friend" is not a good example of the [C]ode FRIEND. Examples should show no more of the context than is needed to see that the [C]ode applies.

Transcription. When coding some kinds of data (for example, video or audio), a researcher has a choice of coding the original data directly, or producing a text transcript before coding. Coding directly from a source like video or audio has several advantages: (1) there is no widely-accepted standard for transcription of the visual content of video data; (2) there is no concern about transcription error; and (3) any transcription is a data reduction, so coding from video or audio data directly preserves context. However, when coding directly from audio or video: (1) segmentation can be more difficult than from transcribed data, and (2) developing automated classifiers, at least with current technology, is difficult. In the ideal case, transcript and original source can be co-registered (for example, using time stamps), so researchers can code from the transcript but preserve the original context. However, we believe that best practice is *not* to code from audio or video data directly but to *add to the transcript information need for coding* (e.g., actions, gestures, intonation). Adding information to the transcript makes the evidence for a [C]ode clear, and thus provides a stronger warrant.

Segmentation. As we argue above, segmentation is a vital part of coding. There is not space here to discuss all of the considerations that go into segmentation, but we do want to make three observations.

Meaningful Breaks. Segmentation is the process of dividing data into *meaningful* pieces. While some researchers segment data by time (e.g., coding 2-s or 1-min intervals of the data), this is not best practice for most contexts. Phenomena of interest can span an arbitrary break in the data, so data should be segmented based on break points that correspond with breaks in the activity being coded: for example, turns of talk in conversation, questions and responses in an interview, sentences or paragraphs in text, or peaks or valleys in a record of galvanic skin response.

Open Coding. One alternative to segmentation is *open coding*, or identifying evidence for a [C]ode in un-segmented data, a practice sometimes used in ethnography. However, it is difficult to achieve good IRR with open coding, and without pre-identified segments of data, it is difficult to account for even simple variables, like duration or frequency of [C]odes. Open coding thus precludes meaningful quantification.

Equivalency. To make quantification possible, segmentation should a divide data into pieces that are exchangeable—that is, equivalent for the purpose of counting.

[C]ode Representation. Researchers also need to decide how to record information about a [C]ode for each segment of data. *Binary representation* is the most common, where a 1 indicates that the [C]ode applies to the segment of data and a 0 indicates that it does not. A *continuous representation* describes a [C]ode by some continuous measure of its intensity or certainty in a segment of data. For example, a value of 0.5 might indicate that the [C]ode applies, but not strongly, or that the rater is 50% confident that the [C]ode applies. Binary representations are more common because people generally have difficulty estimating probabilities of their certainty or intensity of meaning. Some automated classifiers report continuous values, and if a binary and continuous score are being compared, a *cut score* is typically applied: for example, if an automated classifier reports more than 60% certainty, that becomes the equivalent of a human coding positive.

It is possible to construct categorical classifiers: for example, {"[C]ode definitely does not apply," "[C]ode probably does not apply," "[C]ode probably applies," "[C]ode definitely applies"}. These are not recommended because using them usually means converting the categories to a continuous value, and it is difficult to warrant that the intervals between categories are numerically equivalent. Some researchers use a *stacked binary* representation: indicating the binary for each [C]ode using the presence or absence of the label, and listing the labels together. We recommend against this because it is more difficult to conduct IRR or to use quantitative models on a stacked binary representation.

The IRR Loop. The fundamental process of achieving good IRR for a [C]ode involves an iterative cycle of coding, testing, and refining.

Subsets. At any point in the IRR loop, the data to be coded consists of three non-overlapping subsets: (1) A *holdout set*, which consists of all of the segments of data that have not yet been seen by any rater, human or machine, in any way; (2) a *test set*, which consists of any data that has been coded by one or more raters where (a) no rater has seen another's coding, and (b) no test for IRR has been performed; and (3) a *training set*, which consists of any data not in the holdout or test sets.

IRR Principles. IRR can only be warranted by (a) establishing that a test set produces a κ for the whole dataset that is above some threshold, τ, at alpha level, α (typically 0.05 or 0.01), where (b) the test set was sampled from the holdout set. The first principle is discussed above (Sect. 3.1). The second addresses *information leakage* [13, 16]. Coding is a process of *model development*: a mental model for a human rater or a predictive model for an automated classifier. These models are created using the training set and tested using the test set. Information leakage in coding comes about when information about the test set is contained in the training set—that is, when a rater has access to

information about the test set prior to coding. Such leakage can lead to biased estimates of IRR. Thus, the IRR loop works as follows:

1. Rater 1 codes segments of data chosen at random from the holdout set.
2. N items from step 1 are chosen for a test set; remaining items are added to the training set.
3. Rater 2 codes the test set.
4. Values are computed for κ and ρ.
5. If $\kappa(N) > \tau$ and $\rho(\tau) < \alpha$, then we can warrant that κ(full dataset) $> \tau$, and the IRR loop is done; or
6. If $\rho(\tau) \geq \alpha$, then the test fails and the test set becomes part of the training set.
   a. Raters examine the segments of data on which there was disagreement.
   b. If necessary, the definition of the [C]ode is changed.
   c. The raters return to step 1.

Threshold Selection. The choice of threshold depends on the statistic chosen, but historically researchers have chosen thresholds that are lower than is desirable. For example, Cohen argued that κ values of 0.61–0.80 indicate *substantial* agreement [4]. However, raters could disagree on 20% of the data and still achieve $\kappa > 0.60$. It would be difficult to argue that such a [C]ode is fair. To achieve fair [C]odes, researchers should set higher thresholds, as high as $\tau = 0.90$ when using κ.

Eagan and colleagues [5] developed a *binary replicate test* for determining appropriate IRR thresholds. In this approach, once coding is complete and a statistical model has been made, the binary replicate test introduces random error into the coding. The level of κ at which the original result becomes insignificant provides an empirical threshold for acceptable error.

Separability of [C]odes. While there are studies that report an "overall κ" or "κ across all codes," any criteria of fairness has to apply *to each [C]ode individually*. [C]odes can be related to one another; however, it makes no sense to assert that if one [C]ode is manifestly unfair but the rest are fair, that the [C]odes are fair "overall." In applying a [C]ode, we assert that a segment of data *means* something. If that assertion is not defensible, then conclusions drawn from it are not defensible either.

Test Set Size. The optimal size of a test set will vary depending on the IRR statistic being used and the frequency, or *base rate*, of the [C]ode in the data. For κ, we have found that test sets of $80 \leq N \leq 100$ are typically large enough to get acceptable agreement. In early iterations of the coding loop, however, when the raters are just starting to come to a consensus on the meaning of the [C]ode, smaller test sets are more efficient, as raters can more quickly see differences in their understanding.

Test Set Construction. The traditional approach to IRR is to construct a test set by randomly selecting items from the holdout set. (In step 2 of the IRR loop, all of the items from step 1 are included in the test set.) This random selection is necessary for generalizing from the test set to the full set of data. However, researchers who control for Type I error using ρ can also use *conditional sampling*. This is possible because ρ is

an empirical rejective method: it constructs multiple test sets and uses them to determine the likelihood that the actual test set came from data with $\kappa < \tau$.

Specifically, in step 2 of the IRR loop, researchers can select some number, g, of the items from step 1 that Rater 1 coded positive and include those in the test set; the remaining $N - g$ items are chosen at random from the rest of the items coded in step 1. The result is a test set with an *inflated base rate*, or minimum frequency of items that Rater 1 coded positive. This is useful because: (1) more items coded positive provide more information for raters when they disagree; and (2) κ and many other IRR statistics are more sensitive at higher base rates. Taken together, these advantages of conditional sampling—made possible by ρ—make it easier to get high levels of agreement.

Reporting [C]odes. [C] odes should always be described with the *name or label* for the [C]ode, its *definition*, *examples*, and a report of the *results of any IRR* that was conducted. Results of IRR for each [C]ode should include the *statistic* used, the *threshold* chosen, the *statistic value* for the final test set, the *method for controlling for Type I error and its value*, as well as the *size of the test set*. Table 1 shows an example codebook entry for the [C]ode FRIEND.

Table 1 Example codebook entry for the [C]ode FRIEND. IRR columns report Cohen's κ computed on a test set of $N = 80$ items and the result of the ρ test to determine whether $\kappa > \tau = 0.90$. Values are reported for comparison between two human raters (H1/H2) and each human and an automated classifier (H1/AC and H2/AC). Bold entries indicate significance at the $\rho < \alpha = 0.05$ level. *Because κ is not statistically higher than τ for all pairs of raters, the [C]ode is not fair*, and a choice to include such a [C]ode in any analysis would require careful justification.

Code	Definition	Examples	IRR $\kappa(80)/\rho(0.90)$		
			H1 H2	H1 AC	H2 AC
FRIEND	Explicit reference to a being a friend or any related terms (e.g., buddy, pal, chum, bestie), or to the general concept of friendship; could include familial terms in relation to a non-family-member, for example, calling someone "my sister" when they are not actually a sister; referring to a platonic relationship of mutual respect, trust, loyalty, and/or honesty using a single term or phrase	*We're like family, I'll always be there for you* *What are friends for?* *You totally get me, that's why we're BFFs.* *Wuddup cuz?*	0.91 0.06	**1.00** 0.01	**0.96** 0.02

Derived Coding. Thus far, we have been discussing what Shaffer [20] terms *primary [C]odes*. With a primary [C]ode, segments of data can be coded using only evidence from within a segment itself. In contrast, coding a segment of data with a *derived [C]ode*

requires combining two or more pieces of evidence, which may or may not come from the segment being coded. When it is difficult for raters to come to agreement on a complex [C]ode, they can often identify simpler primary [C]odes that can be fairly coded, and then use their combination to make assertions about more complex meaning.

Derived [C]odes are also essential when addressing issues of *context* in coding. Coding a segment of data *in context* means that a decision about the meaning of the segment uses (points to) evidence from elsewhere in the data. As discussed above (see Sect. 2), this act of pointing is always equivalent to some primary [C]ode; thus coding in context can always be accounted for, in principle, using a derived [C]ode.

When using derived [C]odes, it is sometimes effective to use the primary [C]odes directly and let a model of the [D]iscourse connect them into more complex [C]odes. If derived [C]odes are used, however, they need to be validated: it is not the case that if primary [C]odes are fair, any combination of them will be as well.

Automated Coding. There are three types of automated classifiers in use today.

Supervised Machine Learning. Supervised machine learning methods take a set of human-coded segments of data as input. The algorithm identifies a set of *features* in the data (for text data, these could be words, bigrams, trigrams, length of segments, etc.) and then finds the combinations of features that best predict the human coding. Such models use *cross validation* to test the accuracy of the classifiers: the algorithm creates a holdout set and a training set from the human-coded data, then develops a classifier from the training set and tests it on the holdout set. These methods are easy to use; the only human intervention required is the input of the initial coded data. However, supervised machine learning methods require large amounts of data to produce accurate classifiers—often on the order of thousands of segments. Moreover, supervised machine learning models lack transparency: the combination of features is often a complex calculation that cannot be inspected or explained by a human researcher. The warrants for fairness from supervised models may thus appear weaker than other approaches.

Unsupervised Machine Learning. Unsupervised machine learning algorithms, such as *topic modeling*, take a set of *uncoded* text data and extract groups of keywords that are related to one another. These groups can be taken as [c]odes, and human researchers can inspect the keywords to try to discern the [C]odes to which they refer. Unsupervised models are often tempting for researchers to use, especially in exploratory analyses, as they work quickly and require no initial coding. They also provide a transparent model by producing explicit groups of keywords for each [C]ode. However, unsupervised models cannot be used as fair classifiers *unless the implied [C]odes are defined and a complete process of IRR is carried out*. IRR in these circumstances is often problematic because one of the raters (the automated classifier) cannot change its coding. Moreover, from experience we have found that validating unsupervised models requires human raters to code a substantial number of segments of data.

Active Machine Learning. In active machine learning, the algorithm selects segments of data to present to a human rater so as to minimize the number of excerpts that a human rater has to code. For example, *nCoder* [17, 21] automatically constructs test sets with inflated base rates based on its current classifier, including segments that contain

high-frequency words that the human rater has not seen before. When a test set does not result in acceptable IRR, the system shows the segments where raters disagreed and asks the human rater to identify what features they used to make their coding decision. As a result, active machine learning systems can be substantially more efficient than supervised or unsupervised methods. Such systems also represent [c]odes as explicit descriptions of words or patterns of words, and are thus fully transparent.

Advantages of Automated Classifiers. Automated classifiers, once developed, can code large amounts of data quickly and do not suffer from *coding fatigue*: the phenomenon where human raters become less attentive after coding many segments of data. Automated classifiers are consistent and can potentially be applied to new data collected under similar conditions to the original.

Subgroup Fairness. Because many datasets contain subgroups that have low numerical representation in the data—such as populations defined by demographic data such as race or disability status—bias can result when that dataset is used to develop a [C]ode: for example, training facial recognition software using predominantly white faces, and as a result misidentifying the faces of people of color [3, 18]. Many of the methods used to assess fairness depend on the frequency with which events occur in the data. If a subgroup is underrepresented in a dataset, then data from that group may be underrepresented in the test set used to compute IRR and warrant fairness. Attention should be paid to examining the discourse of subgroups relative to any [C]ode. If the code occurs more or less frequently in a subgroup than in the discourse of the community as a whole *without any theoretically-sound reason*, or if disagreements in IRR occur more or less frequently in segments of data from the subgroup, then the coding process should be examined with some care.

3.3 Generating [C]odes

Grounded Generation Cycle. The *grounded generation cycle* is the fundamental process for identifying [C]odes and warranting that the choice of [C]odes is grounded simultaneously in theory, community, and data [2].

1. [C]odes must reflect understanding generated from close reading of the data. Glesne [9] describes coding as a process of sorting and defining, in which a researcher progressively puts similar segments of data together to identify common themes. Shaffer [20] suggests looking for striking or interesting segments of data and using those to identify meaningful categories in the [D]iscourse. In these approaches, the researchers' understanding of the data and the community it represents helps generate a set of fair [C]odes.
2. [C]odes must be fair to some theoretical framework. In qualitative approaches, the researchers' understanding of the setting needs to relate to existing research.

These two perspectives—understanding the data and understanding relevant theory—are sometimes described as *bottom-up* versus *top-down* coding. However, developing good [C]odes is never entirely either a bottom-up or top-down process. Rather, it is an iterative process of moving back and forth between data and theory.

Quantitative Tools. The grounded generation cycle can be supported using quantitative techniques. One common approach is to construct a table of word frequencies that can help researchers check that they have not overlooked some key term of art. Some use *word stemming* to gather together variations on the same word (e.g., command, commands, commander, commanding). Other *natural language processing* approaches can identify words (or N-grams) that occur more frequently in the corpus of data than in common usage, and others (e.g., *latent semantic analysis, topic modeling,* or *skip-grams*) use the frequencies with which words co-occur in common usage to identify families of words that occur together in the data. Any of these approaches are appropriate tools to augment the qualitative process of code generation, *as long as all segments of data that are measured quantitatively are placed in the training set*, and are not used to measure IRR. This includes any summary information about the data, which should exclude information from the holdout set. Thus, a holdout set should be created *at the beginning of the code generation process.*

Automated Approaches to Code Generation. As discussed above, there are algorithms that can identify groups of related words and phrases in uncoded data (topic modeling is one of the best known). Such automated approaches can be useful in identifying potential topics that might be used as [C]odes, and it is possible to use such topics and/or keywords as input to the grounded generation cycle. However, ***there is no way to generate fair [C]odes using only an automated process***. A [C]ode is not only a claim that some topic exists in the data, but that it is a *meaningful* part of the [D]iscourse of some community. This can only be warranted by a process that understands the meaning of segments of data, which existing automated processes cannot do.

Iterativity. This description of creating fair [C]odes might suggest that it is a linear process, however, the iterativity in coding goes beyond the grounded coding cycle and the IRR loop. In practice, the stages of [C]ode identification, implementation, and validation are not independent. For example, the process of getting IRR with one [C]ode might lead to the realization that the definition of the [C]ode needs to change (step 6b in the IRR loop). One persistent question in managing the iterative process of coding is whether to start with broad or narrow [C]odes. We recommend starting with broad [C]odes and progressively making them narrower. It is notoriously hard to get good IRR for very specific [C]odes because fine distinctions between [C]odes are often hard for both human and machine raters to identify. Subdividing an initial broad [C]ode has the advantage of only requiring consideration of segments already identified by the broad [C]ode. Thus, it is generally easier to break a broad code into more specific [C]odes than it is to aggregate very specific [C]odes into a broader category.

Huis Clos. It is sometimes tempting to use existing tools for making claims about data: *sentiment analysis*, for example, or *reading level analysis.* However, whatever their source, and however well established the algorithms that produce them, they are systems for making claims about segments of data. Therefore *all* of the concerns above regarding fairness apply. Consider, for example, a measure like the number of words in a segment of data. This is easy to calculate, but in the context at hand, is the number of words *meaningful*? And if so, *what does it mean*—and *how do we know* what it means?

Thus, any claim about data is a [C]ode, and the process of coding—and concerns about warranting code fairness—are not only universal, but unavoidable.

4 Discussion

We have argued here that coding is a process fundamental to the analysis of data, because coding is the bridge between the world of data and the world of interpretation. The central question in coding, then, is how to make sure that the linkages between data and analysis are valid, in the sense that our interpretation of the analysis and our interpretation of the data are both consistent with one another and fair.

We have consistently addressed this question in terms of *fairness* rather than *validity,* not because quantitative ethnographers can ignore questions of validity, but because we think that the term itself is used in ways that are complex and sometimes contentious. In the end, validity is about meaning. As researchers we care that the meanings we assert respect the communities we are describing, the theoretical constructs we are using, and the data we have collected. In this sense, validity as it is sometimes formally construed is necessary but insufficient. Thus, quantitative ethnographers care not only about cross-validation of coding algorithms, but also about IRR between humans, and between humans and automated classifiers. They care not just about whether coding can be done quickly, but that it is the result of a process that manages [C]ode generation, IRR, coding, and code reporting, all within the framework of best practices for assuring that the results are fair.

Of course, there is room in quantitative ethnography for exploratory analyses. Quantitative ethnographers can, do, and should at times conduct preliminary analyses that use unvalidated codes, or put data into a model as a way of gaining insight into its meaningful structure. It is absolutely possible to start with such approaches and validate them later. But in doing so, we must remember that an exploratory analysis that takes these approaches is, by definition, making claims that may be unfair, and therefore we need to use the utmost caution in using them in any public-facing analysis of data.

We hope that the considerations we have explored above provide a helpful framework for coding data fairly, but we recognize that issues arise in any analysis that cannot be addressed with simple reference to general guidelines. We thus believe that Siebert-Evenstone's Maxim [22] is the best guide for coding in a way that is consistent with theory, community, and the evidence available: *When in doubt, read your data.*

Acknowledgements. This work was funded in part by the National Science Foundation (DRL-1661036, DRL-1713110), the Wisconsin Alumni Research Foundation, and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison. The opinions, findings, and conclusions do not reflect the views of the funding agencies, cooperating institutions, or other individuals.

References

1. Briggs, J.L.: Emotions have many faces: inuit lessons. Anthropologica **42**(2), 157–164 (2000)
2. Charmaz, K.: Constructing Grounded Theory: A Practical Guide through Qualitative Analysis. Sage, Thousand Oaks (2006)
3. Chouldechova, A., Roth, A.: The frontiers of fairness in machine learning. arXiv:1810.08810 (2018)
4. Cohen, J.: Kappa: coefficient of concordance. Educ. Psych. Meas. **20**, 37 (1960)
5. Eagan, B., et al.: The binary replicate test: determining the sensitivity of CSCL models to coding error. In: International Conference on Computer Supported Collaborative Learning (2019)
6. Eagan, B.R. et al.: Can we rely on reliability? Testing the assumptions of inter-rater reliability. In: Smith, B.K. et al. (eds.) Making a difference: Prioritizing equity and access in CSCL: 12th International Conference on Computer-Supported Collaborative Learning. pp. 529–532 (2017)
7. Eagan, B.R., et al.: rhoR: Rho for inter rater reliability (2016)
8. Gee, J.P.: An Introduction to Discourse Analysis: Theory and Method. Routledge, London (1999)
9. Glesne, C.: Becoming Qualitative Researchers: An Introduction. Pearson, Boston (2015)
10. Goodman, N.: Ways of Worldmaking. Hackett, Indianapolis (1978)
11. Herrenkohl, L.R., Cornelius, L.: Investigating elementary students' scientific and historical argumentation. J. Learn. Sci. **22**(3), 413–461 (2013)
12. Hutchby, I., Wooffitt, R.: Conversation analysis. Polity (2008)
13. Kaufman, S., et al.: Leakage in data mining: formulation, detection, and avoidance. ACM Trans. Knowl. Discov. Data (TKDD) **6**(4), 1–21 (2012)
14. Kurasaki, K.S.: Intercoder reliability for validating conclusions drawn from open-ended interview data. Field Methods **12**(3), 179–194 (2000)
15. Lombard, M., et al.: Content analysis in mass communication: assessment and reporting of intercoder reliability. Hum. Commun. Res. **28**(4), 587–604 (2002)
16. Lukács, G., Ansorge, U.: Information leakage in the response time-based concealed information test. Appl. Cogn. Psychol. **33**(6), 1178–1196 (2019)
17. Marquart, C.L. et al.: ncodeR: techniques for automated classifiers [R package] (2018)
18. Mehrabi, N., et al.: A survey on bias and fairness in machine learning. arXiv:1908.09635 (2019)
19. Saldaña, J.: The Coding Manual for Qualitative Researchers. SAGE Publications, Thousand Oaks (2015)
20. Shaffer, D.W.: Quantitative Ethnography. Cathcart Press, Madison (2017)
21. Shaffer, D.W., et al.: The nCoder: A Technique for Improving the Utility of Inter-Rater Reliability Statistics. Epistemic Games Group, Madison (2015)
22. Siebert-Evenstone, A.: Personal communication (n.d.)
23. Thornberg, R., Charmaz, K.: Grounded theory and theoretical coding. In: Flick, U. (ed.) The SAGE Handbook of Qualitative Data Analysis, pp. 153–169 SAGE Publications, London (2014)