When Cohen's Kappa Is Not Enough: Exploring Methods for Estimating Inter-Rater Reliability for Time Sequential Data

Jennifer Czocher
Texas State University

Andrew Baas
Texas State University

Alex White
Texas State University

Kathleen Melhuish
Texas State University

We identify a current pragmatic and methodological problem facing RUME researchers who study processes that unfold over time, like classrooms and interviews, with qualitative coding techniques. In many cases, existing methods for estimating agreement between raters fail to account for the properties of this kind of data, are not compatible with how the coding schemes are applied, and fail to properly estimate rater agreement. Despite many peer-reviewers requesting estimates of agreement for coded data, there is often no suitable value to report and researchers must make do with claims about coming to consensus. We review methods found in the literature and evaluate their suitability, strengths, and weaknesses. While each reviewed method is appropriate for some aspect of this kind of data, none satisfies all desired criteria.

Keywords: methods, quantitative, qualitative, inter-rater reliability

Qualitative researchers in RUME often use coding techniques (e.g. Clarke et al., 2015; Saldaña, 2016) to analyze recordings, transcripts, or artifacts like written work from educational settings like classrooms or interviews. Because qualitative analysis typically adopts a subjective, interpretive approach to seeking patterns in rich, complex data sources, one should not expect two raters would exactly agree on how to apply a coding scheme to a given record of an educational setting. Yet, there is still an expectation that multiple raters trained on the same codebook will conduct an inter-rater reliability (IRR) analysis and report an estimate. Miscalculating and inaccurately reporting IRR has been identified as a common methodological issue when reporting observational data (Hallgren, 2012). In our view, these methodological oversights are symptoms of qualitative researchers not having adequate access to suitable IRR measures or a clear mapping of which measures are suitable for different research settings. In this paper, we discuss the limitations of common approaches to IRR within RUME. We focus on observational data with particular attention to time-sequenced data, to illustrate the impact of differing types of disagreements between raters and consider the viability of existing approaches to estimating IRR. We conclude that the field lacks suitable estimates of IRR for addressing questions about time-sequenced interactions between teachers and students' reasoning, questions of particular interest to the RUME community.

Why Should RUME Care about IRR?

Seeking an objective numerical metric, such as IRR, may appear in conflict with interpretive paradigms of qualitative analysis; however, a suitable IRR method can serve an important role in supporting claims about the qualitative codebook and the coding process. First, if a suitable threshold is achieved, IRR can support inferences about the level of intersubjectivity (stability) of the codebook and therefore increase trustworthiness of the data analysis process. Second, a suitable measure may give information about when a rater has been adequately trained to use a

coding scheme independently. When raters can reliably work independently, the capacity of the research team increases because researchers' confidence that the coding schemes are being consistently applied increases. Relatedly, some researchers intend to disseminate their coding schemes so that other stakeholders, like teachers and school districts, can use them to target and improve qualities of instruction. Third, researchers may wish to track observer drift (Miller, 2018). As raters code independently, they will begin to favor certain codes over others or introduce examples of codes (or additional codes) in distinct ways. It is important to understand how often raters should check in and re-calibrate their usage of the coding scheme. Additionally, raters' convergence or divergence on specific codebooks could become an object of study in itself. Fourth, IRR is essential to mixed methods research, where the resulting frequencies (how much and how often) of code occurrences serve as the input to quantitative methods.

IRR Methods RUME Researchers Use

To situate our discussion within RUME, we conducted a full-text search of the Boston (2022) RUME proceedings and articles published in *IJRUME* using search terms like "agreement" and "rater/coder reliability." The search returned eight proceedings and 16 articles reporting a value for IRR. Only 2 of 8 proceedings and 7 of 16 *IJRUME* articles reported a statistic such as Cohen's κ. A further two articles reported comparable statistics (Fleiss's κ and Krippendorff's α) and two considered correlations between coders. The remaining papers reported an agreement percentage. This brief search suggests that Cohen's κ is the primary method for estimating IRR appearing in published articles in RUME.

What is Cohen's κ and why is it problematic?

Cohen (1960) introduced κ to measure agreement between two raters using a nominal scale to assign analytic units to categories. Cohen's κ (hereafter: κ) is available in many qualitative data analysis software packages. The requirements are fairly stringent: analytic units must be independent, the coding categories must be independent, the codes must be mutually exclusive and exhaustive (MEE), and the raters must be independent. Imagine a recording of a classroom lesson and a codebook containing n different codes, $\{x_i\}$, which do not have a naturally occurring order. The recording is segmented into analytic units. These could be turns of speech or uniform chunks of time, for example. For each analytic unit, Rater 1 can apply x_i (or not) and Rater 2 can apply x_i (or not). For each x_i, and for each segment, there are two agreement states (both apply code x_i, neither apply code x_i) and two disagreement states (one coder applies x_i and the other does not). The counts for each state can be tallied in a contingency matrix and then Cohen's Kappa computes the ratio between (a) the excess probability that two coders agree compared to the probability agreements are due to chance alone and (b) the probability that agreement is not due to chance alone. It is interpreted as the likelihood that raters are coding the same way, corrected for how often the raters (hypothetically) agree by chance alone. Values closer to 1 correspond to stronger inter-rater reliability. Metrics like κ can be suitable for well-structured qualitative data where codes are applied to well-defined analytic units, like turns in a transcript. However, κ rests on quantitative assumptions (e.g., codes are mutually exclusive and represent independent categories) that are not always met by the outputs of qualitative coding.

Four primary categories of observational data can be distinguished: (a) event sequential data encode only the order of events, but not time information about when they occurred, (b) state sequential data assume that every second of the observational record can be mapped to a unique state in the codebook and thus record transitions between states, (c) interval sequential data

consist of equally-segmented intervals and apply codes to each segment, and (d) time sequential data (TSD) allow codable events in the observational record to be identified with a rater-defined time interval (Bakeman & Quera, 1995); codes can be applied to momentary behaviors (e.g., a hand being raised) or to behaviors with a duration (e.g., a teacher engaging in motivational talk). Both the start and stop times of each coded event are relevant in TSD, an aspect which distinguishes it from the other categories and makes it particularly resistant to statistical methods like κ. In TSD data, where one codable event begins is partly constituted by when another one ends (or begins). This means not only that TSD violates independence assumptions, but it also introduces low-level disagreements that researchers may not wish to tally. κ assumes that all types of disagreements between raters are equivalent and should be tallied.

Consider the following three kinds of disagreements: (A) *errors of omission/commission*, where one rater misses (or inserts) a codable event relative to the other rater's analysis, (B) *multiple interpretation problems*, where two raters disagree on what code is appropriate for the analytic unit, but agree that something important relevant to the research question is happening, and (C) *boundary problems*, where two raters agree on what code to use, but choose different times to apply it. The severity of omission/commission depends on whether Rater 1 simply did not perceive the event that Rater 2 conceives as codable, or whether Rater 1 disagrees with Rater 2's interpretation of the event as codable. In the first situation, Rater 1 may simply admit they missed something and adopt Rater 2's code. The second situation should count as a disagreement, but probably not the first. After all, this is a primary reason to have at least two raters reviewing data for mixed-methods work. Multiple interpretations problems will arise when codebooks are detailed and complex. Said plainly, the more choices there are for codes, the more likely two raters will disagree on which code to use even if, at a high level, they agree that a) there is a codable event and b) how to interpret the relevance of the event. Boundary problems are common when using coding schemes that allow for codable events to be selected with a rater-defined time interval instead of uniform time intervals. The raters need to decide when each codable event begins and ends, as well as which code(s) to apply. Each of these types of disagreements is less substantial than two raters fundamentally disagreeing on the nature of what is being observed. The problem is that algorithms vary in their capacity to evaluate what kind of disagreements may have occurred, and therefore whether to count them as "true" disagreements. For these reasons, metrics like κ (and percent agreement) may not properly estimate IRR and thus are limited in their suitability for the kinds of qualitative research common in RUME. In fact, there is growing caution Cohen's kappa should not be considered as a standard for IRR in these cases (Walter et al., 2019).

These problems are exacerbated for projects recording temporally-sequenced observations, like interviews or classroom interactions, and can confront the researcher with distasteful choices. To make the discussion concrete, imagine a cognitive task-based interview setting where the project is interested in studying interviewer-student interactions that lead to successful problem solving. In this example, imagine the research question is about how the interviewer's moves support the student's reasoning. The interviewer's moves might get coded for attributes of the questions she asks or mathematical content of the statements she makes. The student's speech, writing, and gestures might get coded with indicators for phases of problem solving, covariational reasoning, and metacognition – as they occur. These codebooks are very complex and may not be hierarchically organized, thereby resisting simplification efforts.

Due to the nature of mathematical reasoning, multiple codable events may take place during overlapping durations of time. For example, while speaking about a graph, the student may be

speaking about two quantities covarying, but doing so to reflect on the adequacy of results from their previous work (metacognition) by considering an analogous situation (problem solving heuristics). Thus, this TSD does not satisfy assumptions about mutual exclusivity. To meet the assumptions for κ, the observational record would need to be artificially segmented (i.e., transformed into a non-TSD type), which would disregard detailed interactional information the qualitative codebooks are designed to capture. For example, coding the teacher and student moves every 30s would (inaccurately) concede that the only important aspect is whether a move occurred during the time interval, and then apply that code to the entire interval, which would wash away any interactional information during that interval. In short, researchers want to retain "when" something happened as a property of the data so that time-dependent interactions can be studied.

Considerations for Evaluating a Method's Capacity to Exclude Low-Level Disagreements

In general, estimates of IRR are very sensitive not only to raters' consistency in applying codes to the data but also to their accuracy in segmenting transcripts of videos into analytic units. Ideally, the coding process supports assignment of "clearly defined, disjunct categories" (Reed et al., 2018, p. 208) to each codable event and there is an "intersubjective, transparent basis for forming coding units" (Bilandzic et al, 2001 as cited in Reed et al., 2018, p. 208). This is easy enough when the codebook corresponds to direct observations, e.g., *student raises their hand*. It becomes much more complicated with research questions that use interpretive coding schemes, like those associated with student reasoning or teacher scaffolding moves. As discussed above, there are three kinds of low-level disagreements that researchers may not wish to tally. In this section, we review existing methods for estimating IRR for TSD that treat two of them: errors of

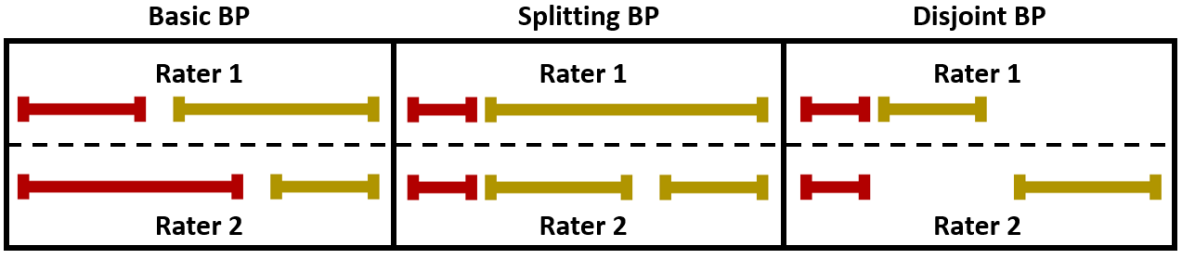

Figure 1 Boundary Problems (BP), which should not count towards disagreements.

omission/commission and boundary problems.

There are three kinds of boundary problems that have implications for the algorithms used to estimate IRR, which we call the basic boundary problem, the splitting boundary problem, and the disjoint boundary problem, diagrammed in Figure 1. The basic boundary problem arises when the raters use non-identical starting or ending boundary points for the codable event. For IRR methods which include some kind of temporal tolerance, a small tolerance can result in false negatives for this kind of boundary disagreement, whereas a large tolerance can result in false positives. The splitting problem occurs when one rater decides to split a code for some non-codable activity that occurred during the interval a codable activity occurs. For example, Rater 1 codes $[t_1, t_2]$ as *quantitative coordination*, but Rater 2 observes that for a duration of time $[t_3, t_4] \subset [t_1, t_2]$ that the student makes a comment about their recent mathematics exam, which is not a codable event, and then goes back to their covariational reasoning activity. Thus Rater 2 produces two codable events, $[t_1, t_3]$ and $[t_4, t_2]$, assigned to *quantitative coordination.* Depending on which IRR method is used, an algorithm could count this as two agreements and

one disagreement or as three disagreements. However, the disagreement is closer in nature to deciding the appropriate duration of an analytic unit, which is why we include it as a boundary problem. A suitable method would be sensitive to the splitting problem only in cases where the number of coded events is important to the research question. The disjoint boundary problem happens when a codable event is a very short duration and is couched in an ongoing student activity. For example, the student may exclaim, "wait!", briefly go silent, and then correct a mistake. Here, the raters may agree on the codable event (*evaluation)*, and which code to apply, but not "when" to give credit for it.

Options for Estimating IRR for Time-Sequenced Data

We sought methods in the literature that aligned with the properties and needs of TSD: (1) they could be modified to accommodate violations of the MEE assumptions (2) the method is properly sensitive to low-level disagreements arising from boundary problems and omission/commission, and (3) the method is adequate to research questions posed about time-dependent interactions. In this section, we review the strengths and weaknesses of three approaches to modifying κ for use with TSD: time-unit kappa, multi-pass, and sequence method. A summary of the methods is provided in Table 1.

Time-unit Kappa. Bakeman et al. (2009) describe the time-unit kappa method as one commonly used in existing IRR literature. This method subdivides the data into very small, uniform, time intervals (typically fractions of a second), and takes the time interval as the analytic unit. As with Cohen's κ, an agreement matrix is formed by counting agreements and disagreements on the subdivisions and κ is calculated on those matrices. As this approach re-defines the analytic units as small time intervals, disagreement between coders on *number* of coded intervals (as in the splitting boundary problem) are not measured. This method is also often modified with a researcher-specified temporal tolerance, allowing it to potentially be insensitive to disjoint boundary problems. The main drawback of time-unit kappa is that it assumes each tally mark in the matrix corresponds to a codable event, thereby dramatically over-estimating the number of rater decisions. Additionally, long strings of subdivisions would all be (apparently) coded identically, violating the independence assumption of κ. Time-unit kappa may still be useful in some situations. In particular, it addresses the splitting boundary problem, can easily be modified to address the disjoint boundary problem, and would be appropriate for research questions that demand high confidence in code location and duration.

Multi-pass. This family of methods improves upon the time-unit kappa method by identifying agreement between complete coded events instead of breaking them up into time segments. This comes closer to meeting the κ independence requirement. They make multiple passes through the coded data and on each pass they assign agreement or disagreement between raters to each coded event which has not yet been assigned a state. There are three published toolkits that execute multi-pass IRR methods: The Observer (Jansen et al., 2003), Interact (Bakeman et al., 2009), and EasyDIag (Holle & Rein, 2015). The Observer makes two passes to tally agreements based on overlap of coded events (within some researcher-defined tolerance) and then makes two more passes to tally code disagreements. The Interact algorithm works similarly but adds an extra pass to identify errors of commission/omission.

Both The Observer and Interact algorithms state that MEE assumptions should be met and both address the three boundary problems. Both are consistent with research questions that care about the occurrence and order of coded events. These algorithms can still be executed on data which violate the MEE assumption, and so potentially may perform well on data which only

minimally violate it (see Jansen et al., 2003). The extent to which they perform well with MEE violations would be testable through simulation methods. However, both algorithms suffer from what Holle and Rein (2015) refer to as the *linking problem*: these methods may produce more entries in the agreement matrix than there are total coded events in the union of the two raters' data. Consider the splitting problem in the middle diagram of Figure 1 from the perspective of Rater 1 and from the perspective of Rater 2. Rater 1's coded event agrees with the first and second events coded by Rater 2 *and* Rater 2's two coded events agree with Rater 1's only coded event. Even though there are only three total coded events, the Observer algorithm would count this as four agreements. Holle and Rein claim that multiple linking can lead to bias towards agreements (2015). Raters who encounter many splitting problems will have artificially inflated reliability.

The EasyDIag algorithm was developed in response to the linking problem in the Observer algorithm (Holle & Rein, 2015). It makes only two passes. In the first pass, it links coded events with sufficient overlap and then removes them from the set. The remaining coded events are considered disagreements. The second pass links the disagreements in order of percent overlap, until the percentage falls below the same threshold. Any remaining coded events are considered errors of omission/commission. While EasyDIag resolves overcounting on the splitting problem, it does not resolve any of the boundary problems. It relies on percent overlap to link coded events and categorize them as (dis)agreements. It would miss boundary problems where the duration of Rater 1's coded event is much longer (or shorter) than Rater 2's coded event. Similarly, it will fail to link coded events with the disjoint boundary problem because it requires some level of overlap between codes. Finally, it will fail to link coded events with the splitting problem because it permits only one link per coded event before removing that coded event from the set of coded events to process. We consider these problems to be potentially resolvable by using a negative threshold, allowing for closely disjoint codes to be associated, which would not be within the range of thresholds (51% to 90%) suggested by the original authors.

Sequence Method. All methods mentioned so far have focused on temporally local agreements between two coders. Bakeman et al. (2009) introduced the GSEQ-DP algorithm which, instead, considers the overall sequence of coded events. This algorithm extends previous work with event sequential data (data viewed as a sequence of coded events, without reference to time stamps). The original algorithm rendered each rater's coded events as a sequence and computed the Levenstein distance between them (Quera et al., 2007). The Levenstein distance measures the minimum number of coded event substitutions, additions, or removals required to transform one sequence of coded events into another. A sequence of transformations is generated in the process and this sequence can be directly mapped to agreements/disagreements (substitutions), commissions (additions), and omissions (removals) in an agreement matrix. κ can then be calculated as usual on the resulting matrix. The GSEQ-DP algorithm connects the sequence-only algorithm to the timing of coded events as follows: Substitutions are free when Rater 1's coded event is temporally close to Rater 2's coded event but a cost is imposed on substitutions that are temporally distant. Imposing the costs ensures the method only identifies agreements between coded events which are temporally close to one another. As with previous methods, the GSEQ-DP algorithm explicitly assume the codebooks satisfies the MEE assumption. Its focus on code sequence resolves the basic and disjoint boundary problems. The splitting boundary problem, though, is still unresolved as the algorithm provides one-to-one mapping between coded events. This method for estimating IRR could be very useful for research questions concerned with the sequencing of coded events, such as, "Does *metacognition*

often follow a teacher's *evaluation of response*?" The algorithm could be amended to relax the MEE requirement and to adequately treat the splitting problem by introducing swapping and splitting as valid transformations, bringing in something closer to Damerau-Levenstein distance instead of the Levenstein distance.

Table 1 Summary of methods for estimating IRR for time-sequence data type: criteria not met (✗), met (✓) and could be adapted (!) for requiring codes be mutually exhaustive and exclusive (MEE), the basic boundary problem (BBP), splitting boundary problem (SBP), disjoint boundary problem (DBP), and commission/omission errors (Com/Om)

Method	No MEE	BBP	SBP	DBP	Com/Om	1:1
Time-Unit Kappa	!	!	✓	!	✓	✗
Observer	!	✓	✓	✓	!	✗
Interact	!	✓	✓	✓	✓	✗
EasyDIag	!	!	!	!	✓	✓
GSEQ-DP	!	✓	!	✓	✓	✓

Conclusions

In this methodological review, we problematized existing methods for estimating inter-rater reliability (IRR) for specific kinds of time-sequenced data common to research programs in RUME. Our discussion distinguished between several kinds of low-level disagreements (errors of omission/commission, multiple interpretation problems, and boundary problems) that can occur when analyzing data qualitatively and what we consider true disagreements – which occur when both raters identify the same event but disagree on which code to apply. True disagreements may have implications for the stability of the coding scheme or speak to the need to re-calibrate the raters. The low-level disagreements should be sorted out through algorithms or quarantined for a reconciliation process between raters. We reviewed Cohen's κ and three methods for extending it for suitability by examining their strengths and weaknesses relative to how they handle the differing kinds of disagreements. While these methods have proven appropriate, and useful, to certain kinds of time-sequenced data, using any of these methods with fidelity would require a shift in research questions, a reduction in complexity of the coding scheme, or modification of procedures for applying the coding scheme.

We agree with Jansen et al. that "how one does reliability analysis depends on what research questions one needs to answer eventually" (2003, p. 394). The metric for estimating IRR should reflect the particularities of how the coding scheme is applied. Unfortunately, many of the measures for IRR make assumptions about the data or coding schemes that do not reflect the kinds of data that the RUME community works with nor the complexity of the coding schemes we develop to study problems of teaching and learning. Instead, researchers are restricted from asking and answering certain kinds of questions because the quantitative tools we have available do not suit the properties of the data. Ultimately, the field needs methods for estimating IRR that respect how the coding schemes are applied to the observational records and thereby comport with the research questions posed.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1750813.

References

Bakeman, R., & Quera, V. (1995). *Analyzing interaction: Sequential analysis with SDIS & GSEQ*. Cambridge University Press.

Bakeman, R., Quera, V., & Gnisci, A. (2009). Observer agreement for timed-event sequential data: A comparison of time-based and event-based algorithms. *Behavior Research Methods*, *41*(1), 137–147. https://doi.org/10.3758/BRM.41.1.137

Clarke, V., Braun, V., & Hayfield, N. (2015). Thematic Analysis. In J. A. Smith (Ed.), *Qualitative psychology: A practical guide to research methods* (3rd edition, pp. 222–248). SAGE Publications.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Hallgren, K. A. (2012). Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial. *Tutorials in Quantitative Methods for Psychology*, *8*(1), 23–34. https://doi.org/10.20982/tqmp.08.1.p023

Holle, H., & Rein, R. (2015). EasyDIAg: A tool for easy determination of interrater agreement. *Behavior Research Methods*, *47*(3), 837–847. https://doi.org/10.3758/s13428-014-0506-7

Jansen, R. G., Wiertz, L. F., Meyer, E. S., & Noldus, L. P. J. J. (2003). Reliability analysis of observational data: Problems, solutions, and software implementation. *Behavior Research Methods, Instruments, & Computers*, *35*(3), 391–399. https://doi.org/10.3758/BF03195516

Miller, S. A. (2018). Measurement (Ch. 2). In *Developmental Research Methods*. SAGE Publications, Inc. https://doi.org/10.4135/9781506331997

Quera, V., Bakeman, R., & Gnisci, A. (2007). Observer agreement for event sequences: Methods and software for sequence alignment and reliability estimates. *Behavior Research Methods*, *39*(1), 39–49. https://doi.org/10.3758/BF03192842

Reed, N., Metzger, Y., Kolbe, M., Zobel, S., & Boos, M. (2018). Unitizing Verbal Interaction Data for Coding: Rules and Reliability. In E. Brauner, M. Boos, & M. Kolbe (Eds.), *The Cambridge Handbook of Group Interaction Analysis* (1st ed., pp. 208–226). Cambridge University Press. https://doi.org/10.1017/9781316286302.012

Saldaña, J. (2016). *The coding manual for qualitative researchers* (3rd edition). SAGE Publications.

Walter, S. R., Dunsmuir, W. T. M., & Westbrook, J. I. (2019). Inter-observer agreement and reliability assessment for observational studies of clinical work. *Journal of Biomedical Informatics*, *100*, 103317. https://doi.org/10.1016/j.jbi.2019.103317