Inter-rater reliability in second language meta-analyses: The case of categorical moderators

Reza Norouzian
The University of Texas at Austin

Abstract

There has recently been a surge of interest in improving the replicability of second language (L2) research (Porte & McManus, 2018). However, less attention is paid to replicability in the context of L2 meta-analyses. I argue that conducting inter-rater reliability (IRR) analyses is a key step toward improving the replicability of L2 meta-analyses. To that end, I first discuss the foundation of IRR in the context of meta-analytic research. Second, I introduce two IRR measures, *S index* and *Specific Agreement*, which aid in improving the replicability of meta-analytic research. Third, I offer a flexible R program, `interrate`, to facilitate the conduct of IRR analyses for L2 meta-analyses. Fourth, I apply our R program to an actual meta-analytic L2 coding sheet to demonstrate the practical use of the IRR methods discussed. Finally, I provide interpretive guidelines to assist both L2 meta-analysts and journals with the transparent reporting of the IRR findings.

Keywords: Replicability, Meta-Analysis, Inter-Rater Reliability, L2 research, Research Methods

Introduction

Meta-analysis has and will very likely continue to gain momentum in second language (L2) research. A key requirement for developing replicable meta-analytic L2 research, however, is that at least two independent L2 experts would agree on the assignment of the study features (i.e., moderators) in their coding scheme to the actual studies collected from the literature. Establishing such *inter-rater reliability* (IRR) has at least two crucial functions. First, it helps to determine the degree to which assignment of codes to studies has resulted from expert judgment rather than occurring by chance (*reliability function*). Second, it allows the meta-analysts to locate the source of their disagreements and modify their code assignments in their coding sheets accordingly (*diagnostic function*). Despite these two critical functions, research (e.g., Belur, Tompson, Thornton, & Simon, 2018; Raffle, 2006) has consistently shown that the use of IRR analyses is largely absent in many systematic research reviews.

Indeed, an inspection of the recent L2 meta-analyses ($N = 34$) published between 2014 and 2019 in 14 L2 journals[1] reveals a similar trend in L2 research. This trend is depicted in Figure 1 (to reproduce Figure 1 see: https://github.com/hkil/m/blob/master/1.r). As can be seen, more than 35% ($n = 12$) of the L2 meta-analyses published in the aforementioned time period either did not report any measure of IRR (i.e., NA) or did not specify what IRR measure they used and what their results were (i.e., Unclear). Nearly 53% ($n = 18$) only relied on the raw agreement percentage (i.e., %Agreement) which has been historically criticized for its inappropriateness (Cohen, 1960; also see next section). And only less than 12% ($n = 4$) of the L2 meta-analyses used Kappa statistic or mixed that with a raw agreement percentage (i.e., Mixed).

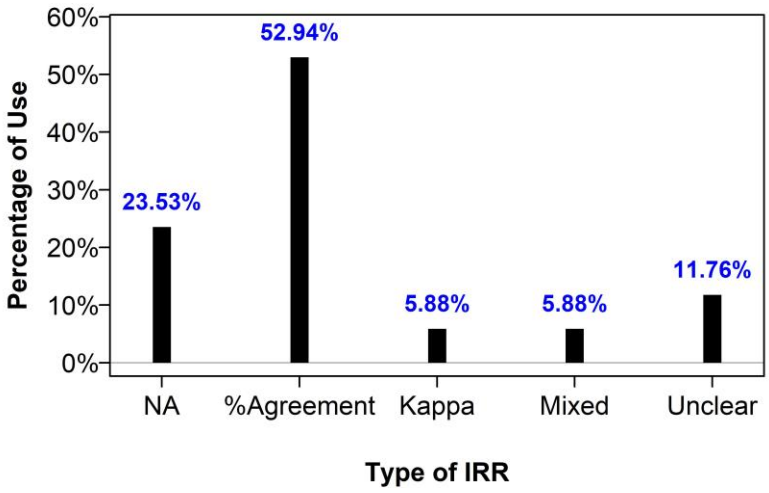

Figure 1. IRR reporting in L2 Meta-Analyses (2014-2019)

In this article, I argue that the limited use of appropriate IRR measures particularly in L2 meta-analyses is rooted in two issues: (a) the paucity of non-technical information regarding the role of IRR in meta-analytic studies and, (b) the practical difficulty in implementing the two aforementioned functions of IRR in meta-analytic research given the number and nature of the categorical moderators employed in L2 meta-analytic research. To respond to these needs, I cover six important areas in this article. First, I discuss the foundation of inter-rater reliability in

the context of meta-analytic research. Second, I introduce two IRR measures (i.e., *S* index and *Specific Agreement*) that fulfill the two aforementioned functions of IRR in meta-analytic research in a non-technical (i.e., without emphasizing formulae or distributional properties of IRR measures) manner. Third, I discuss an IRR measurement difficulty that arises due to the specific nature of categorical moderators used in meta-analysis. Fourth, I offer a flexible R program, `interrate`, that efficiently accommodates the two functions (i.e., reliability and diagnostic) of IRR taking into account the specific nature of each moderator. Fifth, I apply the IRR program to an actual L2 meta-analytic coding sheet to demonstrate the practical use of the IRR methods discussed in the present article. Finally, I provide interpretive guidelines as well as a standard reporting protocol (i.e., appropriate tables and figures) to promote transparent reporting of IRR measures in L2 meta-analyses.

The foundation of inter-rater reliability in meta-analysis

Many L2 researchers appreciate the ability of meta-analytic methods to provide a systematic way of synthesizing a body of L2 research. However, replicability of such research crucially depends on the quality of the human coders' judgment who prepare the meta-analytic coding sheets (see Krippendorff, 2019; Raffle, 2006). The quality of coders' judgment, nonetheless, seems to depend on a host of competing factors. For example, it is expected that as coders code more studies, they make more accurate use of their coding scheme. However, such gains in accuracy may be adversely affected by, for example, a fatigue factor that accumulates over time in the coding process (see Clark, 2008; Lombard, Snyder-Duch, & Bracken, 2002; Rousson, Gasser, & Seifert, 2002). Furthermore, even when coders fully agree on the definition of a study moderator before coding, they may each make different inferences as to how that definition may apply to

each individual study at the time of coding (see Armstrong, Gosling, Weinman, & Marteau, 1997; Cooper, 2017; Krippendorff, 2019).

No matter the source, such differential application of the coding scheme to individual studies could both systematically (e.g., misapplying moderators to studies in consistent ways) and unsystematically (e.g., occasional irregularities in coding) bias the results of meta-analytic research weakening the prospects for its replicability. The extent to which such unreliability might be present in a meta-analytic coding sheet must be closely understood, evaluated, and addressed using appropriate IRR measures (Cooper, 2017).

Appropriate IRR measures in meta-analysis: Reliability function

Coding schemes developed for L2 meta-analyses predominantly involve categorical moderators that are informed by researchers' substantive knowledge as well as the actual realizations of those categories in the published literature. Categorical moderators may also arise in situations where naturally continuous variables (e.g., the time interval between a treatment and a post-test in weeks) may (a) occur in a fairly piece-wise and spread-out fashion (e.g., when sorted: 2, 4, 8, 10, 25, 28 in weeks) in the literature and/or (b) need to be categorized to create theoretically informative benchmarks (e.g., $4 \leq$ short, $5 \leq$ medium ≤ 10, long ≥ 11 in weeks) in any specific domain of L2 research. Either way, because such moderators will eventually serve as predictors of effect size magnitude in a meta-analysis, it is essential that they be reliably applied to each study.

One of the traditional ways to ensure that coders' agreements on a categorical moderator have not arisen simply from random assignment of codes to the collected studies but rather from expert judgment is to use the Kappa statistic pioneered by Cohen (1960) but improved[2] and

generalized to any number of coders by Fleiss (1971). Given the generalizability of Fleiss' Kappa to any number of raters (Fleiss, 1971; Fleiss, Levin, & Paik, 2003, Chapter 18; Hale & Fliess, 1993), throughout the present article, I only use Fleiss' Kappa and for brevity refer to it as Kappa. Succinctly put, Kappa is designed to remove the excess intercoder agreements that may be attributed to chance. How reasonably Kappa can fulfill its reliability function in the context of a meta-analysis, however, is best illustrated using a simple example.

Suppose two L2 meta-analysts have coded ten studies to determine whether their English language learning setting was one of English as a Foreign Language (EFL) or English as a Second Language (ESL). As shown in Table 1, except for two studies, studies 9 and 10, there is a perfect agreement in the coding of this moderator by the two coders.

Table 1. Coding results for two coders

	Moderator (Language Setting)	
Studies	Coder 1	Coder 2
Study 1	EFL	EFL
Study 2	EFL	EFL
Study 3	EFL	EFL
Study 4	EFL	EFL
Study 5	EFL	EFL
Study 6	EFL	EFL
Study 7	ESL	ESL
Study 8	EFL	EFL
Study 9	**ESL**	**EFL**
Study 10	**ESL**	**EFL**

Note. Disagreements are in boldface.

While a raw measure of intercoder agreement would simply assume that there is 80% (8 out of 10 studies) overall agreement, Kappa statistic would also account for a superficial agreement that might have occurred purely by chance between the coders. At this point, to better explore the

Kappa statistic, I suggest using my suite of R functions accessible by running the following in R or Rstudio®:

```
source("https://raw.githubusercontent.com/hkil/m/master/s.r")
```
(R code 1)

The reader should now automatically have access to the coding sheet shown in Table 1 in R or Rstudio® under the name of `table1`. To compute a Kappa statistic for Table 1, we can use the R function `irr`:

```
irr(table1)
```
(R code 2)

The R function returns a Fliess' Kappa of .375 or 37.5% as the chance-free intercoder agreement. But having obtained this result, a question may quickly form in one's mind. Why does the result (i.e., Kappa of 37.5%) indicate such a sharp departure from the raw intercoder agreement (i.e., 80%; 8 out of 10 studies) that naturally exists between the coders in Table 1?

To explore this question, let us suppose instead of Table 1, our two L2 meta-analysts coded their ten studies the way shown in Table 2. The reader should again have immediate access to Table 2 data (i.e., named `table2`) in R.

Table 2. Coding results for two coders

	Moderator (Language Setting)	
Studies	Coder 1	Coder 2
Study 1	ESL	ESL
Study 2	EFL	EFL
Study 3	ESL	ESL
Study 4	ESL	ESL
Study 5	EFL	EFL
Study 6	EFL	EFL
Study 7	ESL	ESL
Study 8	EFL	EFL
Study 9	**ESL**	**EFL**
Study 10	**ESL**	**EFL**

Note. Disagreements are in boldface.

As before, coding of 8 out of 10 studies is in perfect agreement between the coders (i.e., 80% raw agreement). To compute a Kappa statistic for Table 2, we can again use the R function `irr`:

```
irr(table2)
```
(R code 3)

But this time Kappa is estimated to be .6. Indeed, it can be shown that Kappa could still drastically change despite 8 out of 10 studies being always in perfect agreement in the coding sheet. Figure 2 (to reproduce Figure 2 see: https://github.com/hkil/m/blob/master/2.r) visualizes this troubling dynamic for various configurations of our coding sheets in Tables 1 and 2 while keeping 8 out of 10 studies always in agreement.

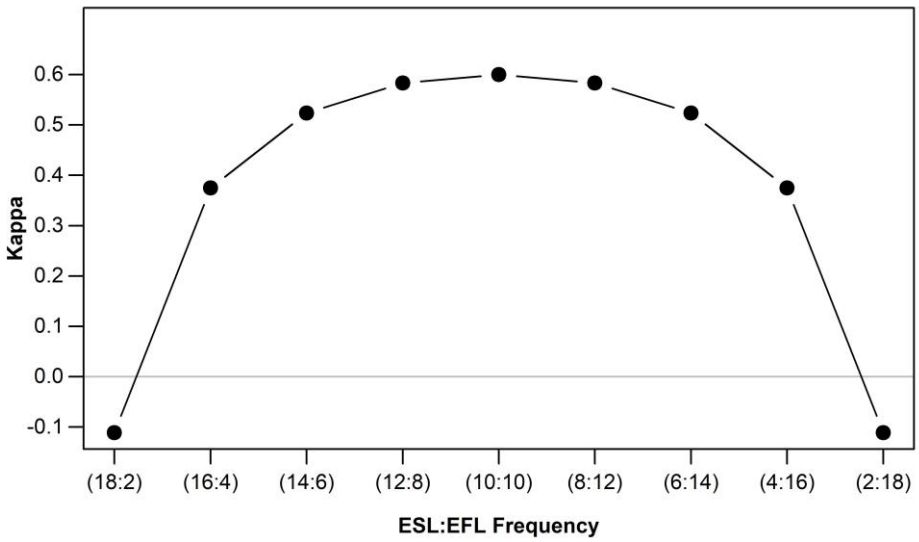

Figure 2. Kappa behavior relative to category frequencies

Put briefly, Kappa is highly sensitive to the distribution of categories of moderators in different coding sheets regardless of the same amount of raw intercoder agreement present in them. Consequently, Kappa for each coding sheet yields a different value. For example, when the overall frequency of EFL category equals that of the ESL category in a coding sheet (each occurring 10 times), Kappa reaches a high of .6. However, as the two categories begin to differ in frequency from one another, Kappa starts to shrink at an alarming rate. In its most paradoxical form, Kappa could even become negative (i.e., real agreements being trumped by those expected by chance) despite 8 out of 10 studies being still in perfect agreement. This is simply because ESL and EFL categories have occurred in a very imbalanced manner (i.e., one occurring 18 times while the other occurring only 2 times) in the coding sheet.

Therefore, despite its popularity, Kappa has a paradoxical behavior (Feinstein & Cicchetti, 1990), and can underestimate the agreement among coders to varying degrees. More recently, an IRR measure to avoid the paradoxes of Kappa, *S* index, has been proposed (Falotico & Quatto,

2010, 2015) which theoretically ranges from –1 to 1 for two coders[3]. Importantly, the *S* index remains unaffected by how categories of a moderator are distributed across different coding sheets with the same amount of raw intercoder agreement. Figure 3 (to reproduce Figure 3 see: https://github.com/hkil/m/blob/master/3.r) shows this desirable feature of *S* index in conjunction with the paradoxical behavior of Kappa.

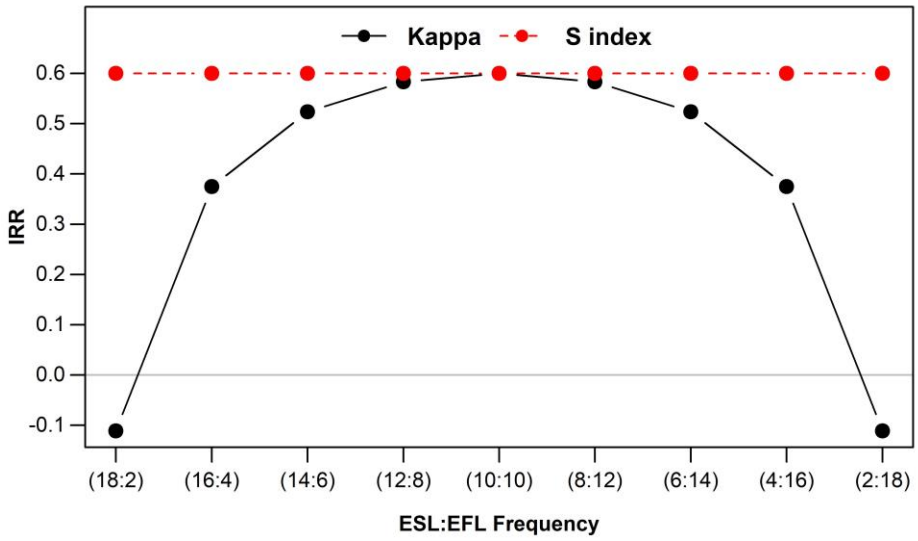

Figure 3. Behavior of Kappa and *S* index relative to category frequencies

As shown in Figure 3, *S* index always yields the same amount of chance-free, intercoder agreement for all configurations of our coding sheets in Tables 1 and 2. As a result, *S* index seems to be a reasonable statistic to fulfill the reliability function of IRR measures.

Appropriate IRR measures in meta-analysis: Diagnostic function

In many meta-analytic situations, when an IRR measure tends to be imperfect or relatively low for a categorical moderator, the next question is which category or categories within that moderator are responsible for the disagreements. For example, if most agreements center on one category while there also are a few disagreements on other categories, L2 meta-analysts will

benefit from knowing what those other categories are to address the source of their disagreements. A diagnostic (i.e., category-specific) version of the Fleiss' Kappa statistic is what is often used in practice (see Gamer, et al., 2019) to perform this diagnostic function (see Fleiss, Levin, & Paik, 2003, Chapter 18). However, this version of Kappa suffers exactly from the same flaw that was discussed in the previous section plus an additional flaw that manifests itself in the case of binary moderators (e.g., EFL vs. ESL). To see both these flaws together, suppose in our running example, one coder coded all ten studies "EFL" with the other coder agreeing throughout expect for the last three cases. The resulting coding sheet is shown in Table 3.

Table 3. Coding results for two coders

	Moderator (Language Setting)	
Studies	Coder 1	Coder 2
Study 1	EFL	EFL
Study 2	EFL	EFL
Study 3	EFL	EFL
Study 4	EFL	EFL
Study 5	EFL	EFL
Study 6	EFL	EFL
Study 7	EFL	EFL
Study 8	**EFL**	**ESL**
Study 9	**EFL**	**ESL**
Study 10	**EFL**	**ESL**

Note. Disagreements are in boldface.

One would expect that a diagnostic measure of IRR would capture the reality that there is a considerable amount of agreement on the "EFL" category (i.e., up to study 7) between the coders while no agreement is observed with regards to the "ESL" category. However, we can use the R function `irr.diag` (i.e., `irr.diag(table3)`) only to find out that the diagnostic Kappa for both "EFL" and "ESL" categories is −0.176. In other words, the diagnostic Kappa is problematic

because it (a) can be negative regardless of the considerable agreements on some categories (i.e., "EFL" in this case) and additionally, (b) is always the same number for both categories which does not even help in locating the low- or no-agreement category (i.e., "ESL" in this case).

To overcome these shortcomings, a more efficient diagnostic IRR measure, *Specific Agreement* (*SA*), was introduced by Cicchetti and Feinstein (1990). As a diagnostic IRR measure, *SA* ranges from 0 to 1 and thus it can be easily expressed in percentages for each category of a moderator. The R function `irr.diag` can be used once again to obtain *SA* for our Table 3 coding sheet:

```
irr.diag(table3)
```
(R code 4)

The function estimates the *SA* to be .824 for the "EFL" category and 0 for the "ESL" category. This result seems to closely match the reality reflected in Table 3. Given that language setting (i.e., EFL vs. ESL) is not often considered a high-inference moderator (see Cooper, 2017, Chapter 4), it is possible that a fatigue factor might have played a role in such a disagreement. But in any case, focusing on why there is no agreement on the "ESL" category would help the two coders make the necessary modifications in their final coding sheet and if needed further clear up possible ambiguities in the definition of this moderator in their coding scheme. As a result, *SA* seems to be a reasonable statistic to fulfill the diagnostic function of IRR measures.

Implementing IRR analysis in L2 meta-analyses

As noted in the *Introduction* section, while developing a foundational understanding of what IRR does is essential, practical difficulty in implementing both functions of IRR in meta-analytic research, nonetheless, remains a formidable impediment to the common use of IRR measures in L2 meta-analyses. In this section, we will be building toward removing this impediment by clarifying the nature of the items coded for measuring IRR indices in meta-analysis.

Nature of the items coded for IRR in meta-analysis: Study-level vs. group-level

Let us operationally define *coding* in the context of IRR. Coding involves assigning a well-defined code (numerical or otherwise) to an item using expert judgment. Importantly, the combination of the terms *item* and *judgment* produces a specific effect in the meta-analytic coding sheets. For example, in Tables 1 through 3 in the previous sections, the moderator *Language Setting* (i.e., EFL vs. ESL) represented a feature that referred to an entire study. Therefore, each study—regardless of how many treatment groups it involved— was overall counted as one item for coding and thus required one count of judgment on the part of each coder. Let us call moderators of this kind *study-level* moderators. But this is not how all moderators appear in a coding sheet. Suppose for example that two L2 coders have examined five L2 writing-related studies to determine whether the writing tasks used in them were timed or untimed (i.e., moderator *Time Restriction*). The coding results are shown in Table 4.

Table 4. Coding results for two coders

		Moderator (Time Restriction)	
Studies	Groups	Coder 1	Coder 2
Study 1	1	timed	timed
Study 1	2	untimed	untimed
Study 2	1	timed	timed
Study 3	1	untimed	untimed
Study 3	2	timed	timed
Study 3	3	untimed	untimed
Study 4	1	timed	timed
Study 4	2	untimed	timed
Study 5	1	untimed	timed
Study 5	2	timed	untimed

Note. Groups denote treatment groups.

In Table 4, each treatment group within a study seems to be capable of falling in a specific category of a moderator in isolation from other treatment groups in that same study. For example, Group 1 in Study 1 seems to have had a timed writing task while Group 2 in Study 1 appears to have used an untimed writing task. Therefore, each study as a whole cannot be counted only as one item for coding requiring one count of judgment. Rather, it is the number of treatment groups within the studies that determine how many items and how many counts of judgment each study presents for coding. Let us call moderators of this kind *group-level* moderators. Understanding this distinction will prove crucial when performing IRR analysis for meta-analysis. Because meta-analytic coding sheets often consist of a mix of both study-level and group-level moderators together, and IRR measures (e.g., *S* index and *SA*) should be able to distinguish between these types of moderators for accurate estimation of intercoder agreement. To illustrate the importance of this dynamic, Table 5 presents the coding results of two coders on two L2 moderators in the same coding sheet.

Table 5. Coding results for two coders on two moderators

			Moderators			
			Time Restriction (*Group-Level*)		Language Setting (*Study-Level*)	
Item	Studies	Groups	Coder 1	Coder 2	Coder 1	Coder 2
1	Study 1	1	timed	timed	ESL	ESL
2	Study 1	2	untimed	untimed	ESL	ESL
3	Study 2	1	timed	timed	EFL	EFL
4	Study 3	1	untimed	untimed	EFL	ESL
5	Study 3	2	timed	timed	EFL	ESL
6	Study 3	3	untimed	untimed	EFL	ESL
7	Study 4	1	timed	timed	EFL	EFL
8	Study 4	2	untimed	timed	EFL	EFL
9	Study 5	1	untimed	timed	EFL	EFL
10	Study 5	2	timed	untimed	EFL	EFL

Note. Item numbers in boldface denote rows formally coded for moderator *Language Setting*.

As noted earlier, moderator *Language Setting* (i.e., EFL vs. ESL) is a study-level moderator, and moderator *Time Restriction* (i.e., timed vs. untimed) is a group-level moderator. As a result, in Table 5, although coders must have used expert judgment on every row of the coding sheet for moderator *Time Restriction* (i.e., 10 items formally coded), for each study as a whole they have used only one count of judgment for moderator *Language Setting* regardless of the number of rows for that study (i.e., 5 items formally coded). Thus, the mere presence of "ESL" and "EFL" codes on every row of the coding sheet (e.g., perhaps by copying from the cell above) does not simply denote that every row has been formally coded using expert judgment (see definition of IRR coding above). In the case of Table 5 coding sheet, if we ignore this fact, the *S* index for moderator *Language Setting* will be .4. However, if we correctly take into account the nature of the moderator *Language Setting*, then the *S* index will actually turn out to be .6. This difference in results occurs because if we disregard the nature of the moderator *Language Setting*, then the disagreements on the moderator's categories in the coding sheet will be incorrectly exaggerated and hence an unjustifiably shrunken IRR measure. Conversely, it is also possible that ignoring the nature of the moderator would lead to inflating or even not changing the IRR measure depending on the distribution of the categories in the coding sheet. However, it is always wiser to avoid such complications in the context of high-stake research such as meta-analysis.

The details provided so far in the present article seem to call for a specialized software program that would possess at least four main capabilities. First, the software should be able to use appropriate IRR measures to fulfill both the reliability and the diagnostic functions discussed earlier. Second, it should be able to perform IRR analyses using one combined or separate coding sheets from any number of coders containing any number of moderators. Third and importantly, it should be capable of distinguishing between the study-level and the group-level

moderators and run the correct IRR analyses separately in each case. Finally, the software should be able to produce a full range of tabular as well as visual information to aid modifying the coding sheet at hand for achieving more replicable meta-analytic research. In the next section, we provide the details of such a software program.

Flexible software for IRR in L2 meta-analysis

To my knowledge, none of the four capabilities stated in the previous section are found in existing reliability packages. For example, the R packages `irr` (Gamer, et al., 2019), and `psych` (Revelle, 2017) do not provide any of the IRR measures (i.e., *S* index and *SA*) discussed in the present article. Additionally, none of the packages can handle meta-analytic coding sheets or distinguish between the moderator types. Other R packages such as `rel` (LoMartire, 2020) either provide a limited version of *S* index suitable only for a two-coder case or like the R package `raters` (Quatto & Ripamonti, 2014) have no power to handle IRR in meta-analytic coding sheets or to provide additional diagnostic IRR measures and graphical aids. Overall, many L2 meta-analysts may find conducting a comprehensive IRR analysis on their coding sheet practically not feasible.

I have developed a flexible, open-source software program in R (R Development Core Team 2019), `intrerrate`, intended to specifically accommodate the goals stated in the previous section and particularly tailored to the needs of L2 meta-analysts with minimal familiarity with R. The software can be freely accessed by the running the first R code (i.e., R code 1) provided earlier. Once the data requirements are correctly met, `intrerrate` performs the full range of IRR analyses discussed in this article for any number of moderators and raters with a single click. In the next section, I will describe these requirements in detail.

Preliminary requirements of `intrerrate`

The R function `intrerrate` requires the Excel sheets containing the coded studies for IRR analysis to have a column named `study.name`. Under this column, coders may consistently select any desired study name (e.g., authors' names). If the exact same author(s) contribute(s) more one than one study to the studies chosen for IRR analyses, the study names must be distinguished from one another accordingly (e.g., Author X_a and Author X_b) under the `study.name` column. Although not required by the program, I also strongly recommend that coders create a second column named based on the group-level feature (e.g., `group.name`) to include the names of treatment groups names or other relevant sub-study names in each study. Doing so allows cross comparability of rows in different coders' coding sheets and further avoids confusion. The function `intrerrate` accepts either a combined Excel sheet containing all coders' coding results side by side (see Table 5) or any combination of separate Excel sheets for different coders. For example, if two coders deliver one combined coding sheet (i.e., side by side) but a third coder presents a separate coding sheet for IRR analyses, all the coding sheets can be fit to the program at once. Also similar to study names, all coders must consistently pick the same moderator names (e.g., 'setting' from coder 1 and 'setting' from coder 2). It is also helpful to know that `intrerrate` has a built-in data-cleaning feature. Therefore, if, for example, some coders habitually allow some blank rows to separate the studies or some blank columns to separate the moderators from one another, they will not need to remove them in their IRR coding sheets as `intrerrate` will do so by design.

When coders intend to input separate Excel sheets to `intrerrate`, it is essential that the rows representing each study's treatment groups (or any other sub-study grouping) have one-on-one correspondence across the coding sheets[4]. For example, if, say, study 4 has two rows

representing its two treatment groups, Y and Z, it is necessary that these two treatment groups be ordered in the same way across all the coding sheets (e.g., from top to bottom first Y, then Z). That said, it would not matter if whole studies, keeping their row orders, appear in different places in the coding sheets for different coders. This is because `intrerrate` locates the studies based on their `study.name`. Thus, no matter where the whole studies appear in each coding sheet across the coders, they will be correctly located by the program. For example, Table 6 provides two coders' coding results that despite their asymmetrical appearance still meet the requirement of `intrerrate`.

Table 6. Two asymmetrical coding sheets for two coders on moderator *Time Restriction*

Order not Important	Order Important	Coder 1	Order not Important	Order Important	Coder 2
Necessary	Recommended	Moderator	Necessary	Recommended	Moderator
study.name	group.name	time.rest	study.name	group.name	time.rest
Study 1	X	timed	**Study 4**	**Y**	**timed**
Study 1	Y	untimed	**Study 4**	**Z**	**timed**
Study 2	X	timed	Study 3	Y	untimed
Study 3	Y	untimed	Study 3	X	timed
Study 3	X	timed	Study 3	Z	untimed
Study 3	Z	untimed	Study 5	X	timed
Study 4	**Y**	**timed**	Study 5	Y	untimed
Study 4	**Z**	**untimed**	Study 1	X	timed
Study 5	Y	untimed	Study 1	Y	untimed
Study 5	Z	timed	Study 2	X	timed

Note. Like all other studies, rows in Study 4 are ordered the same way across the two coding sheets. time.rest = Time Restriction.

Finally, all Excel coding sheets must be saved as CSV Excel files before being input to `intrerrate`. This can be easily achieved via the *Save As* function in Microsoft Excel.

L2 meta-analysis data demonstration with `intrerrate`

It is best to explore the full range of `intrerrate` capabilities by applying it to actual coding sheets for an L2 meta-analysis. I trained a research assistant (RA) in a coding scheme inspired by Kang and Han (2015) for a meta-analytic study focused on the effectiveness of written corrective feedback in developing grammatical accuracy of L2 writers. For the purpose of the present demonstration, my RA and I focused on six moderators whose general, categorical, and coding definitions appear in Table 7.

Table 7. Coding scheme for six selected moderators

Moderator	Abbreviation	General Definition	Categorical Definition	Code
Setting	setting	Language context of the study	Foreign Language Setting / Second Language Setting / Not Available	1 / 2 / NA
Proficiency	prof	Proficiency level of participants	Beginner / Intermediate / Advanced / Not Available	1 / 2 / 3 / NA
Feedback Scope	scope	Range of linguistic structures targeted	Unfocused (> 4 structures) / Mid-focused (2-4 structures) / Highly focused (1 structure) / Not Available	1 / 2 / 3 / NA
Feedback Type	type	Type of feedback provided	Direct (correction given) / Indirect (correction not given) / Meta-linguistic (indirect + grammar notes) / Mixed / Not Available	1 / 2 / 3 / 4 / NA
Error Type	error	Type of linguistic structures targeted	Article / Preposition / Verb / Mixed / Not Available	1 / 2 / 3 / 0 / NA
Random Assignment	random	Random assignment of participants to groups	Yes / No / Not Available	1 / 0 / NA

Note. 'Not Available' denotes that the feature in question was not reported or inferable.

I started the IRR preparation process by selecting a random sample of ten studies from my study pool. This sample included two PhD dissertations and eight journal articles with a total of twenty-one treatment groups studied in them. Also, the sample accounted for nearly 32% of the studies collected for the entire meta-analysis. Next, I prepared a blank Excel sheet with twenty-one rows corresponding to the twenty-one treatment groups in the ten studies as well as two columns named `study.name` (required by `interrate`) and `group.name` (recommended by `interrate`). Then, I populated the `study.name` column using the abbreviated form of the authors' names and the `group.name` column using the abbreviated form of the treatment groups' names. Finally, I shared a copy of this Excel sheet with my RA for coding, provided some clarification regarding how I abbreviated the treatment groups' names, asked her not to change the order of the rows, and discussed other data requirements of `interrate` that were detailed in the previous section.

The reader can have immediate access to the individual as well as the combined coding sheets for my RA and I under the names `c1`, `c2`, and `c3` respectively in R or Rstudio. We can now input either the individual coding sheets (i.e., `c1` and `c2`) or the combined coding sheet (i.e., `c3`) to `interrate` to conduct the full IRR analyses discussed in the previous sections. That is:

```
interrate(c1, c2)
```
(R code 5)

or:

```
interrate(c3)
```
(R code 6)

Understanding the tabular output: Reliability function

In either case, `interrate` will output a range of tabular information similar to what is displayed in Table 8. The variety of information provided in the output mainly serves to assess the reliability function of each moderator coded in the coders' coding sheets.

Table 8. Tabular output of `interrate`

Moderator	*S* index	Lower	Upper	Rows Compared	Minimum Category	Number of Coders	Study Level
error	0.714	0.429	0.929	21	0	2	No
prof	0.867	0.600	1	10	1	2	Yes
random	0.550	0.100	0.854	10	NA	2	Yes
scope	0.857	0.643	1	21	1	2	No
setting	1	1	1	10	--	2	Yes
type	0.492	0.237	0.746	21	3	2	No

Note. Moderators' names are abbreviated per Table 7.

First, the *S* index column lists the intercoder agreements between the two coders on all moderators. Examining this column in our case, we realize that the two coders do not agree much specifically on moderators *Feedback Type* (*S* index = 0.492) and *Random Assignment* (*S* index = 0.55). It is important that, in the context of meta-analysis, any imperfect *S* index (i.e., less than 1) for a moderator be further inspected to reveal the category or categories responsible for the disagreements. In the next section, *Understanding the visual output*, we will learn how to obtain such information for our imperfect IRR measures.

Second, the columns titled *Lower* and *Upper* are the 95% bootstrapped confidence intervals (CI) for the *S* indices which carry important inferential information. Gwet (2014, Chapter 5) argues that these intervals tell us about what we could expect our IRR estimates (e.g., *S* indices) to be, if different coders and items were involved in our IRR coding process over large repetitions (cf.

Norouzian, de Miranda, & Plonsky, 2018, 2019). With a limited number of coders and items, however, these intervals often tend to be fairly wide and uncertain. For example, even though moderator *Random Assignment* currently has an *S* index of 0.55 in Table 8, it is expected that coding different items by different coders could change its current value to anywhere between 0.1 and 0.854.

Third, the column titled *Rows Compared* is where `interrate` looks at the coding patterns of coders on each moderator to ascertain whether that moderator is a study-level moderator or a group-level moderator (see *Nature of items coded for IRR* section). For example, `interrate` has decided that moderator *Proficiency* is a study-level moderator. As a result, it has only counted the ten rows of the coding sheet representing the ten studies not the twenty-one rows representing the twenty-one treatment groups within the ten studies. If we, as human coders, disagree with this decision for legitimate reasons, we can override it using the argument `group.level` resulting in *Proficiency* being analyzed as a group-level moderator:

```
interrate(c3, group.level = "prof")
```
(R code 7)

Fourth, the column titled *Minimum Category* provides the name of the least agreed upon category in each moderator. This nominal information serves to alert the coders to further explore the *SA* indices and locate their specific disagreements (see the next section). Needless to say, when there is perfect agreement among coders (i.e., *S* index = 1), speaking of *Minimum Category* is irrelevant. That is why `interrate` has placed a `--` symbol for moderator *Setting*, because the two coders perfectly agree on all categories of this moderator.

Finally, the titles of the last two columns (i.e., *Number of Coders* and *Study Level*) should reveal what these columns represent. However, I recommend always inspecting these columns to ensure that coders and moderators have been properly picked up and processed by `interrate`.

As a side note, if desired, coders can request an Excel file of the tabular output of `interrate` by providing a `file.name` such as:

```
interrate(c3, file.name = "output")
```
(R code 8)

Now, the coders will have an Excel file named `output` saved in the working directory of their computers containing the output presented in Table 8 which can be transferred over to other documents for formal presentation of the IRR findings. The software will indicate the exact location of the file by generating a message containing the working directory folder name.

Understanding the visual output: Diagnostic function

While the tabular output provides an overall measure of intercoder agreement (i.e., *S* index) for each moderator, understanding what category or categories within each moderator are responsible for the disagreements among the coders can be best explored visually. Figure 4 is automatically generated by `interrate` to serve this purpose.

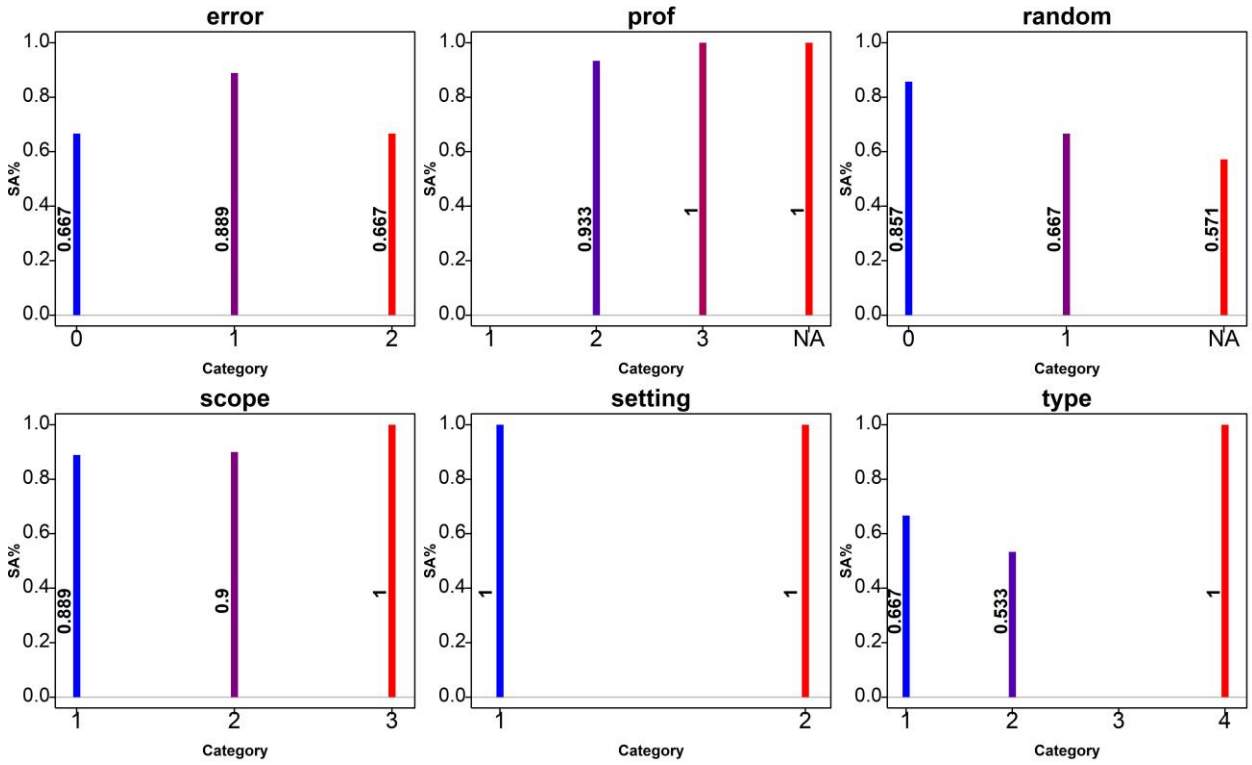

Figure 4. Specific agreement indices for specific categories within six moderators

Let us focus on our two low-IRR moderators from the previous section (i.e., *Feedback Type* and *Random Assignment*). In the case of *Feedback Type*, we see that one of the coders has used "3" (i.e., meta-linguistic feedback) for one or more treatment groups that the other coder has never assigned to any groups (also see Table 3). This is captured by the plot titled *type* not showing any vertical bar for category "3" (i.e., $SA = 0$). Categories "1" (i.e., direct feedback) and "2" (i.e., indirect feedback) also seem not to have been highly agreed upon by the coders ($SA = 0.53$ and 0.66, respectively). However, the perfect agreement (i.e., $SA = 1$) seen on category "4" (i.e., mixed feedback) appears to have helped the overall *S* index for this group-level moderator to be relatively comparable to that for *Random Assignment*.

In the case of *Random Assignment*, we see that the main culprits are the "NA" and "1" categories. This is captured by the plot titled *random* showing lower vertical bars for category

"NA" (*SA* = 0.57) and category "1" (*SA* = 0.66). However, we also see that category "0" (*SA* = 0.84) has seen minor disagreements on perhaps one or two studies as a study-level moderator. Collectively, these disagreements have caused the overall *S* index for *Random Assignment* to be as low as 0.55.

The reader should be able to make similar diagnostic interpretations for the other four moderators in question. During the deliberations following the full understanding of IRR analyses similar to what we showed above, coders can efficiently look for the substantive roots of their disagreements. For example, coders may realize that some of the disagreements in their coding sheets may have occurred purely by mistake. However, other problems may go well beyond being mistakes. For example, curious to know why we agreed so poorly on *Feedback Type*, we realized that the definition of meta-linguistic feedback (coded "3"), as defined in our coding scheme, did not necessarily correspond to all the studies labeling their treatment groups as such. As a result, one of the coders opted for "*Indirect*" feedback, as a cover category, to account for this fact for a handful of treatment groups while the other one elected to follow what the primary authors had labeled "*Meta-Linguistic*" for those groups causing a disagreement. The important point is that `interrate` makes it possible to precisely identify these disagreements and make the necessary changes both in our coding schemes as well as our coding sheets thereby improving the replicability of our meta-analyses.

Another practical problem that might arise when conducting IRR analysis in the context of meta-analyses is disproportionate coding of moderators by different coders. This situation often occurs when coding of studies overlaps in a variety of ways among different coders. In the next section, I explore this issue in more detail.

Disproportionate coding of moderators

In labor-intensive research works such as meta-analysis, it is not uncommon to receive disproportionate contributions from different collaborators. For example, some coders may assist in coding studies only for certain moderators due to their time limitation. Conversely, it is also possible that new moderators are added to the coding scheme and coded by a combination of older and newer coders. In all such cases, a disproportionate number of coded moderators by different coders may be available to the main researcher(s). It will be desirable, if the main researcher(s) could subject all such coding sheets to IRR analyses for more precise estimation of IRR of the moderators that overlap among different coders. By design, `interrate` detects the coders in the coding sheet(s) fit to it, selects the moderators that have been disproportionately coded by different coders and performs the IRR analyses accordingly.

For example, in the case of the L2 meta-analysis example from the previous section, suppose we add two new moderators to the coding scheme shown earlier in Table 7. First, whether each study (i.e., a study-level moderator) included a control group (coded "1") or not (coded "0"). Second, whether participants in each treatment group (i.e., a group-level moderator) had the opportunity to revise their essays after receiving feedback (coded "1") or not (coded "0"). Further suppose that these two new moderators along with two older ones from Table 7 (i.e., *Feedback Type* and *Proficiency*) are now coded by a new research assistant (RA 2) for the same ten studies. As the main researcher, I only need to code the previous ten studies for the two newly added moderators. Naturally, this form of disproportionate coding results in some older moderators (i.e., *Feedback Type* and *Proficiency*) being coded by three coders (the two RAs and I), but others by only two coders (RA 2 and I).

A disproportionate coding sheet akin to what was described above is accessible by running `c4` in R or Rstudio (see online version at: https://github.com/hkil/m/blob/master/c4.csv). This coding sheet consists of all three coders' disproportionate coding of the aforementioned moderators. We can input `c4` to `interrate` as before using:

```
interrate(c4)
```
(R code 9)

It would be instructive to see how the new tabular output shown in Table 9 will differ from the one presented in Table 8.

Table 9. Tabular output of `interrate` for disproportionate coding

Moderator	*S* index	Lower	Upper	Rows Compared	Minimum Category	Number of Coders	Study Level
control	0.8	0.4	1	10	0	2	Yes
error	0.714	0.429	0.929	21	0	2	No
prof	0.911	0.733	1	10	1	*3*	Yes
random	0.55	0.1	0.85	10	NA	2	Yes
revise	0.8	0.4	1	10	0	2	Yes
scope	0.857	0.643	1	21	1	2	No
setting	1	1	1	10	--	2	Yes
type	0.577	0.365	0.788	21	3	*3*	No

Note. Moderators' names are abbreviated per Table 7. Moderator names in boldface are the new moderators. Moderator names in italics are coded by three coders. The moderator underlined must be overridden by user (see text below).

Four changes are observed in the new tabular output. First, notice that `interrate` has properly picked up the number of coders for *Feedback Type* and *Proficiency* moderators as well as the other moderators in a disproportionate way. That is, for some moderators we see two coders while for others we see three coders.

Second, the *S* index for *Proficiency* has now increased from 0.867 to .911 by the addition of a third coder. More important yet is that the 95% CI for *Proficiency* is now narrower (i.e., [0.733,

1]) than what it was in Table 8 (i.e., [0.6, 1]) using two coders. Together, these two improvements indicate that moderator *Proficiency* is going to enjoy a high level of replicability among more coders.

Third, despite the increase in the *S* index for *Feedback Type* from 0.492 to 0.577, the *S* index for this moderator is still low. Interestingly, the least agreed upon category is still category "3" (i.e., meta-linguistic feedback) indicating the need to either modify this moderator in the coding scheme or to add a new category to more clearly distinguish among treatment groups in the collected studies.

Fourth, we see that software has decided, based on the coding pattern of the coders, to treat one of the new moderators, *Revise*, as a study-level moderator. However, as indicated above, this moderator is a group-level moderator whose coding pattern, in this case, happens to resemble that of a study-level moderator in the coding sheet. We can easily override this decision by explicitly designating *Revise* as a group-level moderator:

```
interrate(c4, group.level = "revise")
```
(R code 10)

In this case, overriding `interrate` results in a slight change in the *S* index for *Revise* (i.e., from .8 to .81). As discussed earlier, L2 researchers should recognize the nature of their moderators and make sure that moderators are treated accordingly when conducting IRR analyses for their meta-analyses.

Interpretive guidelines for IRR in meta-analysis

A final practical issue for L2 meta-analysts may relate to interpreting IRR estimates using descriptive benchmarks. While the "choice of such benchmarks . . . is inevitably arbitrary" (Sim

& Wright, 2005, p. 264), to preserve uniformity in interpretations, I recommend the set of benchmarks presented in Table 10.

Table 10. Interpretative guidelines for *S* index and *SA*

	Interpretive Categories				
Measure	Low	Moderate	Acceptable	High	Very Strong
S index / *SA*	Smaller than .6	Between .6 and .7	Between .7 and .8	Between .8 and .9	Larger than .9

These benchmarks are motivated by but more conservative (i.e., set to be higher) than those for Fliess' Kappa for commonly encountered rated item numbers of 25 to 40 (Hale & Fliess, 1993). However, it is harmless for them to be conservative given the possible overreliance on these descriptive qualifiers. Particularly, it must be noted that when only a small portion of a meta-analysis study pool is subjected to IRR analyses, even '*High*' and '*Very Strong*' IRR estimates demand further attention on the part of the coders. Because, it is possible that if two or more coders only marginally disagree on a few studies, their disagreements start to grow as they code other studies. Therefore, the validity of these descriptive qualifiers increases as, among other things, the number of studies used for IRR analyses approaches that of the entire pool of the studies collected for meta-analysis. Stated another way, one should not solely rely on these or any other similar descriptive qualifiers at the expense of ignoring the width of the relevant confidence intervals (see Cumming, & Calin-Jageman, 2017; Norouzian, 2020).

Conclusion

Recently, there has been a surge of interest in improving the replicability of L2 research (see Marsden, Morgan-Short, Thompson, & Abugaber, 2018; Porte, 2012; Porte & McManus, 2018). However, less attention seems to have been paid to the issue of replicability in L2 meta-analyses

which often impact wide-ranging audience in the language learning and teaching world. In this paper, I provided principled solutions to improving replicability in L2 meta-analyses. I also demonstrated these solutions using actual L2 meta-analysis data. Realizing this effort under the methodological reform movement that is currently taking place in L2 research (Gass, Loewen, & Plonsky, 2020; Norouzian, et al., 2018, 2019), I hope that L2 meta-analysts routinely employ the practical methods detailed in the present article. It is also my hope that L2 journals require meta-analytic research submitted for publication to provide the key IRR information needed for replicability purposes. Tables 8 and 9 as well as Figure 4 offer informative and efficient ways to meet that requirement.

Notes

1 The fourteen L2 journals in alphabetical order are: *Applied Linguistics, Applied Psycholinguistics, CALICO Journal, International Review of Applied Linguistics in Language Teaching, Language Learning, Language Learning & Technology, Language Teaching Research, Modern Language Journal, ReCall, Second Language Research, Studies in Second Language Acquisition, Studies in Second Language Learning & Teaching, System,* and *TESOL Quarterly*.

2 Notice the word 'improved'. This means that applying Fleiss' Kappa to only two raters will not result in a Cohen's Kappa. Rather, it can be shown that applying Fleiss' Kappa to only two raters results in a Scott's Phi (Scott, 1955). In general, Scott's Phi is more conservative than Cohen's Kappa.

3 In general, *S* index ranges from $-\left[\frac{1}{\text{Number of Coders}-1}\right]$ to 1. Therefore, as the number of coders increases, the lower bound of *S* index approaches 0.

4 Unlike the study names, the names of the sub-study column (e.g., `group.name`) in meta-analyses very frequently co-occur. For example, in an L2 meta-analysis on the corrective feedback, many treatment groups may be similarly named 'focused'. As such, no software can use sub-study features to re-organize a coding sheet.

References

Armstrong, D., Gosling, A., Weinman, J., & Marteau, T. (1997). The place of inter-rater reliability in qualitative research: an empirical study. *Sociology, 31*, 597-606.

Belur, J., Tompson, L., Thornton, A., & Simon, M. (2018). Interrater reliability in systematic review methodology: Exploring variation in coder decision-making. *Sociological Methods & Research, 0*, 1-29.

Cicchetti, D. V., & Feinstein, A. R. (1990). High agreement but low kappa: II. Resolving the paradoxes. *Journal of Clinical Epidemiology, 43*, 551-558.

Clark, T. (2008). We're over-researched here! Exploring accounts of research fatigue within qualitative research engagements. *Sociology, 42*, 953-970.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*, 37-46.

Cooper, H. (2017). *Research synthesis and meta-analysis: A step-by-step approach* (5th ed.). Thousand Oaks, CA: Sage Publications.

Cumming, G., & Calin-Jageman, R. (2017). *Introduction to the new statistics: estimation, open science, and beyond*. Chicago: Routledge.

Falotico, R., & Quatto, P. (2010). On avoiding paradoxes in assessing inter-rater agreement. *Italian Journal of Applied Statistics, 22*, 151-160.

Falotico, R., & Quatto, P. (2015). Fleiss' kappa statistic without paradoxes. *Quality & Quantity, 49*, 463-470.

Feinstein, A. R., & Cicchetti, D. V. (1990). High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology, 43*, 543-549.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*, 378-382.

Fleiss, J.L., Levin, B., & Paik, M.C. (2003). *Statistical Methods for Rates and Proportions, 3rd Edition. New York*: John Wiley & Sons.

Gamer, M., Lemon, J., Fellows, I., & Singh, P. (2019). irr: Various Coefficients of Interrater Reliability and Agreement. R package version 0.84.1. Retrieved from https://CRAN.R-project.org/package=irr

Gass, S., Loewen, S., & Plonsky, L. (2020). Coming of age: the past, present, and future of quantitative SLA research. *Language Teaching.* Advance online publication. https://doi.org/10.1017/S0261444819000430

Gwet, K. L. (2014). *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters* (4th ed.). Gaithersburg, MD: Advanced Analytics.

Krippendorff, K. (2019). *Content analysis: An introduction to its methodology* (4th ed.). Thousand Oaks, CA: Sage publications.

LoMartire, R. (2020). rel: Reliability Coefficients. R package version 1.4.2. Retrieved from https://CRAN.R-project.org/package=rel

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

Marsden, E., Morgan-Short, K., Thompson, S., & Abugaber, D. (2018). Replication in second language research: Narrative and systematic reviews and recommendations for the field. *Language Learning, 68*, 321-391.

Norouzian, R. (2020). Sample size planning in quantitative L2 research: A pragmatic approach. *Studies in Second Language Acquisition*. Advance online publication. https://doi.org/10.1017/S0272263120000017

Norouzian, R., De Miranda, M., & Plonsky, L. (2018). The Bayesian revolution in second language research: An applied approach. *Language Learning, 68*, 1032 – 1075.

Norouzian, R., De Miranda, M., & Plonsky, L. (2019). A Bayesian approach to measuring evidence in L2 research: An empirical investigation. *Modern Language Journal, 103*, 248 – 261.

Porte, G. (2012). *Replication research in applied linguistics*. New York, NY: Cambridge University Press.

Porte, G., & McManus, K. (2018). *Doing replication research in applied linguistics* (2nd ed.). NY, New York: Routledge.

Quatto & Ripamonti (2014). raters: A modification of Fleiss' Kappa in the case of nominal and ordinal variables. R package version 2.0.1. Retrierved from https://CRAN.R-project.org/package=raters

Raffle, H. (2006). *Assessment and reporting of intercoder reliability in published meta-analyses related to preschool through Grade 12 education* (Unpublished doctoral dissertation), Ohio University.

R Development Core Team. (2020). R: A language and environment for statistical computing: R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/

Revelle, W. (2017). psych: Procedures for Personality and Psychological Research. Retrieved from https://CRAN.R-project.org/package=psych

Rousson, V., Gasser, T., and Seifert, B. (2002). Assessing intrarater, interrater and test–retest reliability of continuous measurements. *Statistics in Medicine, 21*, 3431-3446.

Scott, W. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly, 19(3),* 321-325.