Journal of Data Science xx (xx), 1–24 DOI: xx.xxxx/xxxxxxxxx
July 2020

Improving the Science of Annotation for Natural Language Processing

Kylie Anglin*[1], Arielle Boguslav[2], and Todd Hall[2]

[1]Department of Educational Psychology, University of Connecticut, Storrs, CT, USA

[2]Department of Education Leadership Foundations and Policy, University of Virginia, Charlottesville, VA, 22902

Abstract

Text classification has allowed researchers to analyze natural language data at a previously impossible scale. However, a text classifier is only as valid as the the annotations on which it was trained. Further, the cost of training a classifier depends on annotators' ability to quickly and accurately apply the coding scheme to each text. Thus, researchers need guidance on how to generate training data with optimal efficiency and accuracy. To this end, this study proposes the single-case study design as a feasible and causally-valid research design for empirical decision-making in annotation projects. The key strength of the design is its ability to generate causal evidence with as few as one annotator. In this paper, we demonstrate the application of the single-case study in an applied experiment and argue that future researchers should incorporate the design into the pilot stage of annotation projects so that, over time, a causally-valid body of knowledge regarding the best annotation techniques is built.

Keywords *annotation; coding; NLP; text classification; single-case study.*

1 Introduction

Text documents provide a rich source of data for linguists and social scientists alike. As these researchers bring their analyses to scale, text classification is playing an increasingly important role across many research domains. Without text classification, traditional

*Corresponding author. Email: kylie.anglin@uconn.edu

Todd Hall was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305B200005 to the University of Virginia.

approaches to analyzing natural language data rely on highly trained personnel to read and annotate each document of interest. First, researchers use theory to define a *coding scheme*: a systematic framework for labeling each document that is tailored to the specific research question (Shaffer and Ruis, 2021). Second, researchers either apply those labels to the documents themselves or hire annotators to read and code each text. In addition to participating in extensive training processes, hired annotators are often required to have relevant professional and educational experiences that relate to the project's specific research area. The coding process is therefore both time and resource-intensive, limiting the number of documents that can be included in any given study.

Text classification methods, on the other hand, only require hand-labeling text for a subset of the documents comprising the training data. These data are then used to train an algorithm to automatically apply the coding scheme to the remaining documents in the corpus. Notably, once the text classification algorithm has been trained, it can be applied again and again to additional documents at negligible cost. This feature makes text classification a powerful and efficient analytic tool when study populations are large and the amount of text data is voluminous. However, the validity and cost of text classification depends on annotators' ability to apply the coding scheme accurately and efficiently to each text.

Unfortunately, researchers can currently find only limited guidance on how to produce valid and efficient hand-labelled training data. While qualitative social scientists have devoted substantial attention to methods of producing valid human codes (Creswell and Miller, 2000), this field has not traditionally needed to be concerned with the efficiency of the coding procedure, nor whether the labelled text, and the complex social constructs represented by those labels, are appropriate for automatic text classification. For more specific advice on producing training data for machine learning, researchers may instead turn to the field of computational linguistics where researchers have recently begun to build a "science of annotation", advising researchers on the best methods of producing training data (Hovy and Lavid, 2010). However, this science of annotation is still in a nascent stage. Despite growing calls for researchers to document the origins and appropriate uses of training data in data statements or data sheets (Gebru et al., 2021; Bender and Friedman, 2018), many papers today still fail to report key information on how their training data were obtained (Geiger et al., 2020). Further, while researchers can find insightful and practical recommendations (Hovy and Lavid, 2010; Ide and Pustejovsky, 2017; Pustejovsky and Stubbs, 2012), there are still few studies that have empirically tested the best methods of annotation. A few key exceptions include research on the potential influence of annotator characteristics (Alyuz et al., 2021; Snow et al., 2008), of iterative consensus building among annotators (D'Mello, 2016), and of pre-annotation (Lingren et al., 2014), as well as several reviews of annotation software (Dipper et al., 2004; Neves and Ševa, 2021; Neves and Leser, 2014).

Currently, empirical tests of annotation techniques commonly take one of two forms; both of which create challenges for identifying the causal effect of particular annotation methods. In one common approach, researchers may use a pre-post design where annotators use one method of annotation followed by an alternative method. Performance statistics are then compared across the time points. This design is straight-forward but presents severe challenges for causal inference. Namely, it is impossible to decipher whether changes in annotator performance are due to the new method of annotation, or due to increased annotator experience or any of a number of other time-varying confounders (Shadish et al., 2002). In another approach, researchers may split annotators into two groups and ask each group to use a different annotation method. Performance statistics are then compared across groups. However, in this design, the causal impact of the annotation design cannot be differentiated from differences in performance due to annotator characteristics. This is particularly problematic when there is a small number of annotators and/or they have not been randomly assigned to their annotation condition (Shadish et al., 2002).

In this paper, we demonstrate that the single-case study design can be a key method for building and improving the science of annotation for social scientists and computational linguists alike. This design addresses both time-varying and participant-varying sources of confounding variables by switching the annotation procedure multiple times and comparing outcomes within (rather than across) participants. If the annotation procedure is manipulated many times by the researchers, and the changes in performance track this pattern of manipulation, the researcher can conclude a causal relationship. A key strength of the single case study design is that it can be used with as few as one annotator (Kratochwill et al., 2013). The strong causal validity and low participant requirements make it well-suited to empirically testing the efficacy of various annotation techniques in projects relying on just a handful of annotators. For this reason, we argue that researchers should use the single-case study design to guide decisions during the pilot phase of an annotation project and share the results of those studies, increasing the body of empirical knowledge in annotation.

This article proceeds as follows. First, we provide an overview of the single case study design for those who may not be familiar. Second, we review key decision points in annotation projects, highlighting points where the single-case study can aid in empirical decision making. Third, we illustrate the application of the design through an applied experiment testing two competing approaching to multi-label annotation projects. Finally, we discuss the generalizability of our results and the strengths and weaknesses of the single-case study design for improving annotation science.

2 The Single-Case Study Design

Single-case study designs have a long history in psychology, dating back to the field's founders (Perone and Hursh, 2013; Skinner, 1938; Watson, 1925). In contrast to the between-subject design, the single-case study relies on within-subject comparisons, where the participants provide their own control data. The researcher assigns different treatment conditions to the same individual at different points in time while consistently measuring the outcome of interest. If the treatment assignment is manipulated many times by the researcher and the changes in outcomes track this pattern of treatment manipulation, the researcher concludes that the treatment caused the changes in outcomes. This conclusion is warranted when it is difficult to hypothesize confounders that would also produce the observed pattern of effects (Kratochwill et al., 2013). Conclusions from a single case study are primarily drawn from visual analysis of graphs (Kratochwill et al., 2010). To provide evidence of a treatment effect, the graph should demonstrate an unlikely change in the pattern of data that correlates with the researcher's manipulation of the treatment condition (What Works Clearinghouse, 2019). A stylized example of a convincing single case study is provided in Figure 1.

According to the What Works Clearinghouse (a governmental organization that rates the rigor of empirical evidence in education), the single case study design is one of only three designs (including the randomized control trial and the regression discontinuity design) which meet high standards for causal evidence (2019). A strong single case study has the following features: 1) the treatment is manipulated by the researcher, not by the study participants or the environment; 2) the outcome variables are measured systematically and consistently over time; and 3) there are at least three switches in conditions. Together, these conditions reduce the likelihood of confounding variables that produce the same pattern of effects as the manipulation in the treatment assignment.

The single case study gets its name from the fact that the design can include as few as one participant. This feature makes it attractive for determining the impact of interventions when the participant pool is small. For example, the design is particularly popular in areas of psychology focused on evaluating treatments for rare or low-incidence diagnoses (Carbone et al., 2013). The limited sample size requirements also make it a low-cost yet causally valid design for researchers making decisions at the beginning of a large annotation project. Annotation projects often include only a handful of annotators, making other causally valid designs, like the randomized control trial, infeasible.

The key weakness of the single case design is its potentially limited generalizability. While results can provide evidence of a causal effect for a single individual, this effect may or may not generalize beyond that individual. For this reason, researchers are expected to provide a comprehensive description of participants so that readers may consider the extent to which the impact of an intervention is likely to generalize to their population of interest (Kratochwill et al., 2013). Replicating the single case study design with multiple participants can also provide stronger evidence of a generalizable effect. Further, there are scenarios when a researcher is most interested in identifying a causal effect for their own participants, without any need for generalization. This may occur in clinical cases when an individualized treatment must be chosen, or in annotation projects where the researcher wishes to choose the the most efficient method of annotation for their specific set of annotators. In these cases, the single-case study has few disadvantages.

[Figure 1 about here.]

3 Key Questions in the Science of Annotation

Annotation is a complex and multi-part process. As a result, researchers are faced with many decisions in designing and implementing a coding scheme. Here, we focus on two key decisions in an annotation project which can be answered empirically: Who should create the annotations? And how should they do it? For the most part, we sidestep the question of *what* should be annotated, as that decision is wholly dependent on the research question at hand. We'll simply note there is broad agreement that 1) the annotated corpus needs to be representative of the population of interest (Manning and Schütze, 1999); and 2) that researchers should create a comprehensive codebook which specifies the definitions of codes and provides examples (Hovy and Lavid, 2010). Because codes need to be theoretically valid and appropriate for the data at hand, creating the codebook is often an iterative process where the researcher moves back and forth between theory and data before finalizing the code definitions (Auerbach and Silverstein, 2003; Hovy and Lavid, 2010). Helpfully, researchers can find substantial guidance on creating a codebook from the literature on qualitative research. See, for example, Auerbach and Silverstein (2003), Chi (1997), and Shaffer and Ruis (2021).

3.1 Who should annotate?

One of the first decisions researchers need to make in an annotation project is who should create the annotations. Researchers may produce the annotations themselves, identify content-area experts to produce the annotations, train undergraduate or graduate students (as is common in academic papers), or rely on untrained annotators from crowd-sourcing platforms like Amazon's Mechanical Turk (Geiger et al., 2020). There is a general understanding that the cost per annotation resulting from crowd-sourcing can be substantially less expensive than the cost per annotation resulting from content-area experts (Snow et al., 2008; Fort, 2016). However, it is also hypothesized that crowd-sourced annotations will be of lower quality. This hypothesis has been, at least partially, substantiated with empirical evidence. In a comparison of annotations created by expert annotators to those created by crowd-sourced workers, Snow et al. found higher agreement among expert annotators than between expert and non-expert annotators. However, they also found that accuracy can be increased to the level of that achieved by experts by aggregating the annotations of multiple non-experts (2008). Importantly, the accuracy costs of relying on non-expert annotators will be very dependent on the task at hand. The above tests, for example, were completed on tasks requiring only general knowledge of the English language. More specialized tasks may result in lower accuracy among non-experts. Where relevant, researchers may test this in their own data. Thankfully, there are few causal challenges in identifying the effect of one group of annotators versus another. This is because when testing the impact of different annotators, the researcher does not need to worry about annotator characteristics confounding the outcomes; differences between annotators are not confounders but instead the treatment of interest. Thus, so long as the researcher holds other features constant (like time and the annotation task), comparisons of outcomes across participant pools is valid.

3.2 How should the corpus be annotated?

After determining the annotators, researchers need to determine how the annotators will produce their annotations. This involves selecting the annotation procedures and the annotation interface. Regarding the annotation procedures, researchers need to make two key decisions. First, if the annotation task involves multiple codes, should annotators annotate one code at a time, or all at once? Second, if the annotation task involves long documents, what amount of context should annotators use to interpret each text segment?

Social science annotators are often advised to work through one document at a time and to consider each piece of text within the context of the full document (Shaffer and Ruis, 2021). Computational linguists, on the other hand, are commonly advised to break a complex annotation project down into a series of simple micro-tasks: asking annotators to consider one code at a time and to view the text within just a small context window (Sabou et al., 2014; Hovy and Lavid, 2010). We label these two competing approaches to annotation the *complex* and *simple* annotation schemes.

Computational linguists commonly argue that the simple annotation scheme can increase efficiency by placing a lower cognitive load on annotators (Hovy and Lavid, 2010; Ide and Pustejovsky, 2017; Sabou et al., 2014). Hovy and Lavid argue, for example, that "though [the simple annotation procedure] compromises on sentence context, [it] is both far quicker and far more reliable: annotators need to hold in mind just one set of alternatives, and become astonishingly rapid and accurate" (2010, p. 10). Similarly, the makers of the popular new annotation software, Prodigy, celebrate the software for allowing annotators to "focus on one task at a time" (Explosion AI, 2017). Though there is an accuracy cost to removing an utterance from its context (Samei et al., 2014), doing so also allows researchers to simplify the annotation task, which is hypothesized to increase annotator efficiency and accuracy enough to make up for performance lost due to lack of context. Further, by decomposing a task into simple yes or no questions, it becomes more feasible to rely on untrained annotators through crowd-sourcing for large-scale projects (Sabou et al., 2014). For example, this is the approach of the Decompositional Semantics Initiative, which decomposes complex linguistic concepts into "straightforward questions on binary properties that are easily answered" by untrained native speakers (White et al., 2016, p.1713). Breaking a multi-label task into multiple simple questions also has the added benefit of flexibility: when annotators code for all codes at once, the codebook becomes brittle. Changes to the coding scheme would require re-annotating all utterances. The simple approach allows for codes to be changed or edited without wasting substantial effort (White et al., 2019). However, this simple approach to annotation is rarely taken by social scientists, either in traditional qualitative research or in text classification. In social science projects, annotators commonly consider a document at a time, annotating for every code in the codebook at once, increasing the cognitive load but also increasing the information available to annotators (see, for example, D'Angelo et al., 2020; Loksa and Ko, 2016). The applied experiment in this paper demonstrates how the single case study design may be used assess the trade-offs between these two perspectives while controlling for participant and time-varying confounders.

After specifying the annotation procedures, researchers need to identify the annotation interface, i.e., the software with which the annotators will interact. Ide and Pustejovsky identify many potential interfaces including asking annotators to maintain a simple comma-separated-value file, contribute to a SQL database, or use a software specifically designed for annotation (2017). Neves and Ševa also provide an extensive review of annotation software based on technical criteria (including the cost and easiness of installation), data criteria (including the input and output format of documents), and functionality criteria (including whether the software supports multi-label annotations and document-level annotations). Following these criteria, they recommend three programs that likely meet the needs of most users: WebAnno, brat, and FLAT. Unfortunately, however, there is currently little causally-valid evidence comparing the accuracy and efficiency of annotations resulting from competing interfaces. Helpfully, the single-case study design provides a key opportunity to affordably obtain such information.

4 Applied Experiment

In this section, we demonstrate how the single case study design may be used to inform the development of annotation projects and to answer key questions in annotation science. Specifically, we empirically assess two competing approaches to human annotation. In the simple approach, the annotation task is broken down into short and simple micro-tasks: annotators view short text segments while considering one coding category at a time. In the complex approach, annotators consider all codes at once and consecutively annotate text segments within a full document.

The study is situated within a broader educational research project focused on the efficacy of one-on-one coaching for improving teacher practice. The goal of the research is to use text classification to automatically monitor the strategies employed by coaches in their conversations with teachers and teachers-in-training. To this end, a coaching expert developed a coding scheme and codebook by iteratively drawing on coaching research, practitioner resources, their professional experience receiving and providing coaching, and a random sample of coaching transcripts. The initial coding scheme included over 30 potential strategies. For the purposes of text classification, we will initially focus on eight of the most common strategies. These include: positive evaluation, observation, suggestion, instruction, demonstration, anticipation, practice, and encouragement. A description of these strategies, along with examples, are provided in Table 1. In a single turn, a coach can employ as many as eight strategies or as few as zero. This means our project involves a multi-label classification task in which there are multiple categories (distinguishing it from a binary classification task) and many can apply at once (distinguishing it from a multi-class task).

[Table 1 about here.]

4.1 Study Corpus and Participants

Our corpus of coaching conversations come from prior studies of the impact of a short (5-minute) coaching intervention on teachers-in-training. For more details on the coaching intervention and its effects, see Cohen et al. (2020). All coaching conversations were recorded, professionally transcribed, and segmented by turns of talk. Here, we randomly selected 30 coaching transcripts for piloting annotation, 508 utterances in total. Then, we developed a gold-standard corpus; two coaching experts read the randomly selected transcripts and carefully labelled each coach utterance with the appropriate codes (agreement = 0.96, Krippendorff's alpha = 0.82). Because accuracy was the only priority in the creation of the gold standard corpus, the experts viewed each utterance within the context of the full transcript and took no steps to increase their own efficiency.

Four annotators were recruited through the university's centralized system for hiring undergraduate workers. The job was advertised to students across all schools and majors at the university. Applicants submitted a resume and short cover letter explaining their interest in the project and participated in a short video interview. While all four annotators had research experience and were in their third or fourth year of study, only two had prior teaching experience or a major within the school of education. Three out of four annotators had prior experience with qualitative coding, specifically. This hiring and recruitment process followed the typical approach in social science research projects (Crittenden and Hill, 1971).

In a follow-up experiment exploring mechanisms (discussed later in detail), we sampled an additional 20 transcripts, 360 utterances in total. This follow-up experiment was conducted with three of the four annotators. (One annotator graduated from the university and could not participate in the follow-up experiment.)

4.2 Annotation Procedures and Interface

In line with our annotators' prior technological experiences, we chose a simple annotation interface implemented in Excel. Utterances were displayed in one column of the interface and the annotators would enter their codes in a separate column (or columns). Specific annotation instructions depended on whether the annotators were coding under the complex or simple annotation scheme.

Under the complex annotation procedure, annotators were asked to code one transcript at a time and to consider all coaching strategies at once. To this end, their coding interface included one file per transcript. In each file, transcripts were formatted so that each row was a turn-of-talk. Turns-of-talk were kept in the order in which they were spoken, including both coach speech and teacher-in-training speech. For each coach utterance, annotators would select codes from a drop down menu containing the eight coaching strategies and an option for "None of the above." When appropriate, annotators could select multiple codes. When annotators finished coding a transcript, they would open the next file to continue coding the next transcript. For an example of this annotation interface, see Figure 2.

In the simple annotation scheme, annotators were asked to code for one coaching strategy at a time. Thus, annotators were provided with one file per code (rather than one file per transcript). Again, each row was a turn-of-talk. However, turns-of-talk were presented in random order so that utterances were viewed with only the preceding teacher turn of talk as context. Annotators were then asked to enter a zero or one indicating whether the coach's speech was an exemplar of the target code. Once annotators finished coding all utterances for one coaching strategy, they would open the next file and code the same utterances for the next code. For an example of this annotation interface, see Figure 3.

All utterances were coded four times, with two annotators using the complex annotation scheme and two annotators using the simple annotation scheme.

[Figure 2 about here.]

[Figure 3 about here.]

4.3 Measures

For each annotation procedure, we developed analogous methods for measuring efficiency and validity. To assess annotator efficiency, annotators were asked to record their start and end time for each coding file (either the time it took to annotate a transcript or the time it took to annotate all potential exemplars of a coaching strategy). We then converted these values into a measure of time spent per utterance-code, which served as our efficiency metric. In the complex annotation scheme, this was simply the average time it took an annotator to consider the appropriate codes for a coach turn-of-talk. In the simple annotation scheme, this was the summation of the average time it took annotators to consider an utterance for each of the eight coding tasks. Because the simple scheme requires coding the same utterance multiple times (here, eight times), a full picture of coding efficiency requires us to calculate total time spent coding per utterance.

To assess validity, we measured the accuracy, precision, and recall of the resulting annotations under each procedure. Accuracy here is defined as annotator agreement with the gold-standard corpus. We measured accuracy by calculating the percent of correctly classified utterance-code pairs; because the annotators classified each turn-of-talk as representative – or not – of eight separate codes, it was possible for an annotator to accurately classify an utterance for one code, but incorrectly classify the utterance for a second code. Because our transcripts were imbalanced (all codes are present in less than 50% of utterances, and some are present in less than 10%), it is also important to measure precision and recall. We measure precision by calculating the proportion of true positive utterance-code pairs out of all positively coded utterances and measure recall by calculating the proportion of true positive utterance-code pairs that the annotator identified as such.

4.4 Study Design

We first randomly assigned four annotators to their starting condition (either the simple or complex annotation procedure). After the first week of coding, annotators were instructed to switch their method of annotation (from the simple to the complex, or vice versa) at the beginning of each of the successive three weeks of coding (see Table 2). In single case study terms, this design is referred to as the ABAB design. It is the switching mechanisms that provides the study with high causal validity; if the impact of switching conditions is strong enough, then it is very difficult to hypothesize alternative explanations for the observed changes in outcomes. Thus, if the simple annotation scheme greatly increases (or decreases) annotation accuracy or efficiency, these changes can be causally attributed to the annotation condition. Our study design meets all of the What Works Clearinghouse standards for a causally-valid single case study (What Works Clearinghouse, 2019).

[Table 2 about here.]

4.5 Statistical Analysis

In this study, efficiency is calculated each week for four weeks and four annotators, resulting 16 data points in total, an insufficient number of observations for statistical tests of significance, particularly given the dependency structure. However, we have over a thousand annotations for each participant: enough to determine whether, for each participant, there is a statistically significant difference in precision, recall, or accuracy depending on the annotation condition. To this end, we estimate

$$\begin{aligned} Y_{ijk} = \beta_1 Simple_{ik} Annotator1_{ik} + \beta_2 Simple_{ik} Annotator2_{ik} + \\ \beta_3 Simple_{ik} Annotator3_{ik} + \beta_4 Simple_{ik} Annotator4_{ik} + \\ Week_i \beta + Annotator_k \beta + Code_j \beta + \epsilon_{ijk}, \end{aligned} \tag{1}$$

where Y_{ijk} is a binary variable for whether a given turn-of-talk, i, was accurately coded for code j, by Annotator k. $Annotator$ is a vector of indicators for each of the four annotators, $Week$ a vector of indicators for each of the four weeks, and $Code$ a vector of indicators for each of the eight codes in the coding scheme. The coefficients of interest here are β_1 through β_4, the average impact of the simple coding procedure for each of the four annotators.

Statistical analyses can also be helpful for summarizing results across participants, though readers should be careful not to misinterpret the tests of significance. Statistical inference here is used to make inferences from a sample of utterances to a population of utterances, not from a sample of annotators to a population of annotators. We summarize the results across our four participants

$$Y_{ijk} = \beta_1 Simple_{ik} + \beta_2 Week_i + Annotator_k \beta + Code_j \beta + \epsilon_{ijk}, \tag{2}$$

where Y_{ijk} is a binary variable for whether a given turn-of-talk, i, was accurately coded for code j, for annotator k. $Annotator$ is a vector of indicators for each of the four annotators, $Week$ a continuous variable for the week, and $Code$ a vector of indicators each of the eight codes. The coefficient of interest here is β_1, the average impact of the simple coding scheme across all four annotators and eight codes.

All models were estimated using the `statsmodels` (Seabold and Perktold, 2010) and `pandas` (Wes McKinney, 2010) packages in `Python 3.10.0` (Van Rossum and Drake, 2009). Figures were produced using `matplotlib` (Hunter, 2007) and `seaborn` (Waskom, 2021).

4.6 Results

The effect of the simple annotation procedure on annotator efficiency is presented in Figure 4. The figure demonstrates that the simple coding procedure took roughly twice as long as the complex coding procedure to produce the same number of codes. In the complex procedure, annotating an utterance for all of the eight codes at once took 35.5 seconds on average. Annotating an utterance for a single code took only 8.5 seconds, but this approach requires annotators to read each utterance eight separate times, thus, requiring 68 seconds per utterance to produce the same number of codes as the simple procedure (the sum of the average time spent on each of the eight individual codes). In other words, though reviewing an utterance for a single code took annotators less time than reviewing the utterance for multiple codes, the time spent was not reduced by a factor of eight, which would be required to make the simple annotation more efficient than the complex procedure in this case.

From a single case study point of view, Figure 4 provide convincing evidence of causality; manipulation of the treatment condition here is associated with a consistent change in the dependent variable. The effect is visually obvious at each switch in the treatment conditions and is replicated for participant in the study. We see that each individual takes more time when coding under the simple annotation procedure than when coding under the complex annotation procedure. For one individual, this effect seems to be small (Annotator 2), while for the others it is much larger. Crucially, it is very difficult to provide any alternative explanation for the change in times given that the effect is demonstrated at every switch in treatment condition and for every annotator. No other confounding variable is likely to display this same pattern of effects.

[Figure 4 about here.]

Unlike Figure 4, Figure 5 does not demonstrate a strong consistent impact of the simple annotation procedure on accuracy. While the simple annotation procedure causes a decrease in accuracy for one annotator (Annotator 2), the effect is not convincingly replicated with the other annotators. In Table 3, we summarize the average accuracy for each annotator under the two annotation schemes using Equation 1. While annotator accuracy was high across the board (over 95% for each annotator), there are no statistically significant differences in accuracy by annotation procedure for any of the four annotators. When we aggregate these results across annotators using Equation 2, the overall impact of the simple annotation scheme is a small, but significant, decrease in accuracy by half a percentage point.

[Table 3 about here.]

[Figure 5 about here.]

Given the imbalanced nature of our data set, an analysis of precision and recall is important for understanding the impact of the simple annotation procedure. Figure 6 demonstrates that the simple annotation procedure caused a decrease in precision for three out of four annotators: on average, a statistically significant negative effect of 3.5 percentage points across all four annotators. On the other hand, Figure 7 demonstrates a heterogeneous impact of the simple annotation scheme on recall. While two annotators experienced substantial and consistent impacts of the simple annotation procedure, these effects are in opposite directions (-12 percentage points for Annotator 2 and +10 percentage points for Annotator 3; see Table 3). These two effects counterbalance one-another, resulting in a very small, non-significant, effect for recall overall. Taken together, the simple annotation procedure increases the time spent coding and reduces precision, but only has a minimal negative impact on overall accuracy.

[Figure 6 about here.]

[Figure 7 about here.]

4.7 Follow-up Experiment to Determine Mechanisms

Compared to the complex annotation procedure, the simple annotation procedure is different in two keys ways: 1) the simple procedure provides annotators with less context surrounding an utterance; and 2) the simple procedure asks annotators to review an utterance for a single code at a time. The previous results demonstrated that the simple annotation procedure is less efficient and results in annotations with a lower rate of precision. A natural follow-up question is whether this decrease in efficiency and precision is due to the increase in context or to considering a single code at a time. To test this, we conducted a short follow-up experiment which only varies the context provided to annotators. In both conditions, annotators code for all labels at once (as in the complex procedures), but in one condition ("in-context"), annotators view the utterance within the context of the full transcript. In the other condition ("out-of-context"), annotators only view the preceding utterance. As in the previous design, annotators switched conditions each week.

[Table 4 about here.]

Figure 8 demonstrates that there is no substantial or consistent efficiency difference for either condition. Thus, context was mostly irrelevant in determining the amount of time coders take to produce annotation. There is also no consistent impact for accuracy or precision. However, when coding in-context, annotators produce annotations with higher recall (by four percentage points; see Figure 9 and Table 4). Taken together with the results of the prior experiment, the results suggest that the amount of context was unimportant for determining annotator efficiency and that the increase in efficiency of the complex annotation procedure was due to annotators considering all codes at once. Interpreting the results of the two experiments in conjunction is a little more complex when considering accuracy, precision, and recall. While the simple annotation procedure reduced precision, a lack of context reduced recall. We discuss potential explanations for these results below.

[Figure 8 about here.]

[Figure 9 about here.]

5 Interpreting the Results of the Applied Experiments

The above applied experiments tested two key questions in the design of a multi-label annotation task with long documents: should annotators annotate one code at a time, or all at once? And, what amount of context should annotators use to interpret each text segment? Given the results above, we have determined that the best procedure for the annotators in this study is to annotate for all codes at once within the context of a full document (i.e., the complex annotation procedure). We did not find any benefits to accuracy or efficiency resulting from the simple annotation procedure. In total, it took annotators twice as long to code the same data using the simple annotation procedure than using the complex annotation procedure. Whatever cognitive speed was gained by requiring annotators to only consider one code at a time was not enough to outweigh the time it takes to consider the same utterance multiple times.

When considering precision and recall, our results are a bit more complex. While in the main experiment, the simple annotation procedure reduced precision, in the follow-up experiment, a lack of context reduced recall. How can we explain the seemingly distinct results? Consider that recall is a measure of the relationship between the number of false negatives and true positives. If annotators identify fewer true positive, recall will be reduced. It is unsurprising then that reducing context decreases recall because, without context, annotators may not be able to identify every relevant application of a code. On the other hand, annotating for one code at a time likely increases the number of true positives because coders are forced to consider the applicability of every code. Thus, these two opposing mechanisms cancel one-another out in the complex procedure. The remaining decreasing in accuracy in the simple procedure, then, is due to reduced precision: by forcing annotators to consider each code, they are nudged towards (falsely) believing a code is applicable.

Of course, readers should be thoughtful in their consideration of whether the findings of this study generalize to their own context. In particular, there are two dimensions along which generalizability should be considered. First, our study was conducted with undergraduate research assistants, three of whom had prior experience with annotation in other qualitative studies across the university. We might hypothesize that reducing cognitive load is more important when annotators lack experience or knowledge of the study context. While our annotators were not content area experts, they were also not novices to the same degree as annotators hired through crowd-sourcing platforms like Amazon Mechanical Turk. However, we believe this project has optimistic implications for the use of crowd-sourcing for social science text classification projects. Crowd-sourcing platforms *necessitate* short simple annotation tasks. MTurkers, for example, expect each task to take a matter of seconds (Sabou et al., 2014). Previous research has demonstrated that crowd-sourced annotators can compete with the accuracy of more traditional annotators (Snow et al., 2008), however, this research does not address the potential loss of accuracy that comes with altering the annotation task so that it may be crowd-sourced. This study demonstrates that while simplifying an annotation task and taking excerpts outside of their larger context may reduce accuracy slightly, it is not to such a degree that social science researchers need to dismiss crowd-sourcing as a possibility. Second, the relative trade-offs of the simple and complex annotation procedures are likely to depend on the coding scheme itself. In particular, the benefits of the simple annotation scheme is likely to vary by the number of codes in the codebook, though the function of this relationship is unclear. Further, the amount of text context required for sufficient accuracy will depend on the constructs in the coding scheme. Codes which depend on information provided earlier in conversation will necessitate large context windows. We suggest that in cases where any of these conditions are meaningfully different from the current study, that researchers conduct their own tests. A key strength of the single-case study design is that such tests can be completed quickly and a relatively low cost.

6 Conclusion

This study demonstrates a straight-forward and low-cost method of testing hypotheses regarding the design of annotation projects: the single case study design. Given the limited number of participants required to make causal inferences, the single case study design is well-suited to answer annotation questions when projects have only a few annotators.

While the randomized control trial would be preferable in the case where an annotation project includes many annotators (say, close to 30), in our experience, researchers rarely hire that many annotators outside the context of crowd-sourcing. The single case study design, on the other hand, is valid with as few as one annotator. Thus, researchers can pilot annotation procedures quickly and cheaply, while also obtaining findings with high causal validity. Though each single case study may only generalize to a subset of annotation projects, the relatively low cost of the design means that replicating findings across various contexts is feasible. Thus, we encourage researchers to use the single case study both to inform their own annotation projects and to iteratively improve the evidence base regarding best practices in human annotation.

In the past, many text classification papers have neglected to give human annotations the consideration they are due (Geiger et al., 2020). Indeed, human-annotated training data have been given so little attention that researchers have deemed human-annotated corpora, the "hidden pillars of the domain" (Fort, 2016, p. 9). Thankfully, there is a growing literature on annotation, as evidenced by key texts like Hovy and Lavid (2010) and Pustejovsky and Stubbs (2012), but there are still few empirical tests of the impact of annotation conditions on annotation efficiency and quality. We argue, therefore, that researchers should respond to calls for increased attention to annotation quality by incorporating causal evidence into decision-making when designing annotation projects. If human annotations are the "hidden pillars" of text classification, we believe that we can increase the strength and visibility of these pillars through an increased focused on empirical, causally-valid, decision making in annotation.

Supplementary Material

The results of this paper may be reproduced using the scripts and data files uploaded to Harvard Dataverse.

References

Alyuz N, Aslan S, D'Mello SK, Nachman L, Esme AA (2021). Annotating Student Engagement Across Grades 1–12: Associations with Demographics and Expressivity. In: *Lecture Notes in Computer Science*, 42–51. Springer International Publishing.

Auerbach C, Silverstein LB (2003). *Qualitative Data: An Introduction to Coding and Analysis*, volume 21. NYU press.

Bender EM, Friedman B (2018). Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6: 587–604.

Carbone VJ, O'Brien L, Sweeney-Kerwin EJ, Albert KM (2013). Teaching Eye Contact to Children with Autism: A Conceptual Analysis and Single Case Study. *Education and Treatment of Children*, 36(2): 139–159.

Chi MT (1997). Quantifying Qualitative Analyses of Verbal Data: A Practical Guide. *Journal of the Learning Sciences*, 6(3): 271–315.

Cohen J, Wong V, Krishnamachari A, Berlin R (2020). Teacher Coaching in a Simulated Environment. *Educational Evaluation and Policy Analysis*, 42(2): 208–231.

Creswell JW, Miller DL (2000). Determining Validity in Qualitative Inquiry. *Theory Into Practice*, 39(3): 124–130.

Crittenden KS, Hill RJ (1971). Coding Reliability and Validity of Interview Data. *American Sociological Review*, 36(6): 1073.

D'Angelo ALD, Ruis AR, Collier W, Shaffer DW, Pugh CM (2020). Evaluating How Residents Talk and What it Means for Surgical Performance in the Simulation Lab. *The American Journal of Surgery*, 220(1): 37–43.

Dipper S, Götze M, Stede M (2004). Simple Annotation Tools for Complex Annotation Tasks: An Evaluation. In: *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, 54–62.

D'Mello S (2016). On the Influence of an Iterative Affect Annotation Approach on Inter-Observer and Self-Observer Reliability. *IEEE Transactions on Affective Computing*, 7(2): 136–149.

Explosion AI (2017). Prodigy: A new tool for radically efficient machine teaching. https://explosion.ai/blog/prodigy-annotation-tool-active-learning.

Fort K (2016). *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*. John Wiley & Sons.

Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, Iii HD, et al. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Geiger RS, Yu K, Yang Y, Dai M, Qiu J, Tang R, et al. (2020). Garbage in, Garbage out? In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM.

Hovy E, Lavid J (2010). Towards a 'Science' of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. *International Journal of Translation*, 22(1): 25.

Hunter JD (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95.

Ide N, Pustejovsky J (2017). *Handbook of Linguistic Annotation*, volume 1. Springer.

Kratochwill TR, Hitchcock J, Horner RH, Levin JR, Odom SL, Rindskopf DM, et al. (2010). Single-Case Designs Technical Documentation. *Technical report*, What Works Clearinghouse.

Kratochwill TR, Hitchcock JH, Horner RH, Levin JR, Odom SL, Rindskopf DM, et al. (2013). Single-Case Intervention Research Design Standards. *Remedial and Special Education*, 34(1): 26–38.

Lingren T, Deleger L, Molnar K, Zhai H, Meinzen-Derr J, Kaiser M, et al. (2014). Evaluating the Impact of Pre-Annotation on Annotation Speed and Potential Bias: Natural Language Processing Gold Standard Development for Clinical Named Entity Recognition in Clinical Trial Announcements. *Journal of the American Medical Informatics Association*, 21(3): 406–413.

Loksa D, Ko AJ (2016). The Role of Self-Regulation in Programming Problem Solving Process and Success. In: *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ACM.

Manning CD, Schütze H (1999). *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge.

Neves M, Leser U (2014). A Survey on Annotation Tools for the Biomedical Literature. *Briefings in Bioinformatics*, 15(2): 327–340.

Neves M, Ševa J (2021). An Extensive Review of Tools for Manual Annotation of Documents. *Briefings in Bioinformatics*, 22(1): 146–163.

Perone M, Hursh DE (2013). Single-case experimental designs.

Pustejovsky J, Stubbs A (2012). *Natural Language Annotation for Machine Learning: A Guide to Corpus-Building for Applications.* O'Reilly Media, Inc.

Sabou M, Bontcheva K, Derczynski L, Scharl A (2014). Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. In: *In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA) (2014) 859–866.*

Samei B, Olney AM, Kelly S, Nystrand M, D'Mello S, Blanchard N, et al. (2014). Domain Independent Assessment of Dialogic Properties of Classroom Discourse. In: *Proceedings of the 7th International Conference on Educational Data Mining*, 4.

Seabold S, Perktold J (2010). statsmodels: Econometric and statistical modeling with python. In: *9th Python in Science Conference.*

Shadish WR, Cook TD, Campbell DT (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference.* Houghton Mifflin.

Shaffer DW, Ruis AR (2021). How We Code. In: *ICQE 2020* (S Lee, AR Ruis, eds.). Springer, Malibu, CA.

Skinner BF (1938). *The behavior of organisms: An experimental analysis.* BF Skinner Foundation.

Snow R, O'Connor B, Jurafsky D, Ng A (2008). Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 254–263. Association for Computational Linguistics, Honolulu, Hawaii.

Van Rossum G, Drake FL (2009). *Python 3 Reference Manual.* CreateSpace, Scotts

Valley, CA.

Waskom ML (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60): 3021.

Watson JB (1925). Experimental studies on the growth of the emotions. *The Pedagogical seminary and journal of genetic psychology*, 32(2): 328–348.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In: *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt, Jarrod Millman, eds.), 56 – 61.

What Works Clearinghouse (2019). What Works Clearinghouse Standards Handbook: Version 4. *U.S. Department of Education's Institute of Education Sciences (IES)*, 1–17.

White AS, Reisinger D, Sakaguchi K, Vieira T, Zhang S, Rudinger R, et al. (2016). Universal decompositional semantics on universal dependencies. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1713–1723.

White AS, Stengel-Eskin E, Vashishtha S, Govindarajan V, Reisinger DA, Vieira T, et al. (2019). The universal decompositional semantics dataset and decomp toolkit. *arXiv preprint arXiv:1909.13851*.

List of Figures

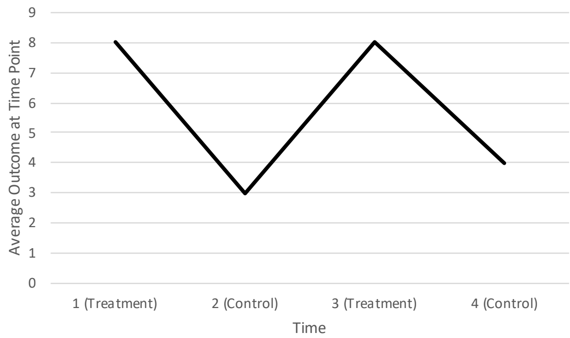

Figure 1: Stylized example of ABAB single case study design with one participant and clear causal impact. Outcomes may either be single observations taken from the participant or average outcomes across many observations of the same participant. The strongest single case studies are also replicated multiple times with more than one participant.

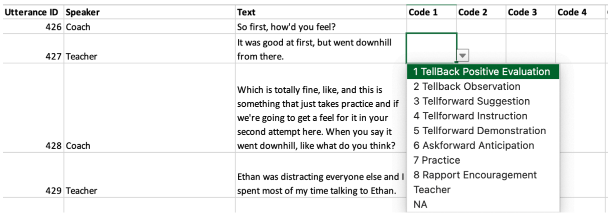

Figure 2: Complex annotation procedure interface. All coach and teacher utterances in a given transcript were included in the order in which they were spoken. Annotators considered one transcript at a time.

STRATEGY: Positive Evaluation			
Utterance ID	**Preceding Teacher Text**	**Coach Text**	**Code**
603	Mm-hmm.	The other thing I do want to point out is there were quite a few times throughout your past stimulation when um you did provide various specific um instructions for an attempt to redirect the misbehavior.	0
416	I think I did a good job of like asking students to explain or like pullback in the text to explain their answer instead of just what they think.	I heard you say, “what in the text make you think that?” or like “could you read me this part of the text?” and that’s like a great probe for like textual evidence, that was awesome. Are there things that you, like, think you could have done differently in terms of feedback?	1

Figure 3: Simple annotation procedure interface. Coach utterances were presented in randomized order along with the preceding teacher utterance. Annotators considered one code at a time.

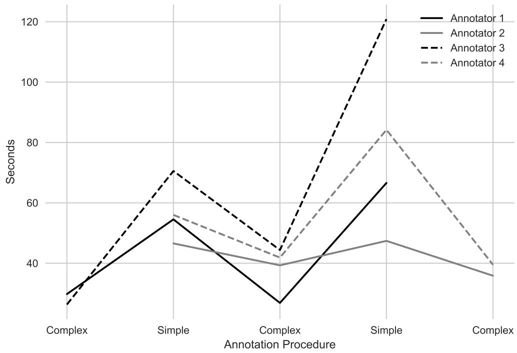

Figure 4: Average total annotation time per utterance as a function of the annotation procedure. The y-axis of the figure displays the total average annotation time per utterance. Under the complex annotation scheme, this is the average time it takes the annotators to consider the relevance of the eight codes all at once for a given utterance. Under the simple annotation scheme, this is the average total time it takes annotators to consider the relevance of the eight individual codes one at a time.

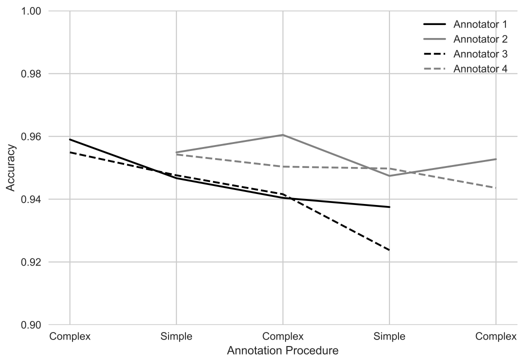

Figure 5: Accuracy as a function of the simple versus complex annotation procedure. The y-axis of the figure displays average accuracy across all utterance-code pairs.

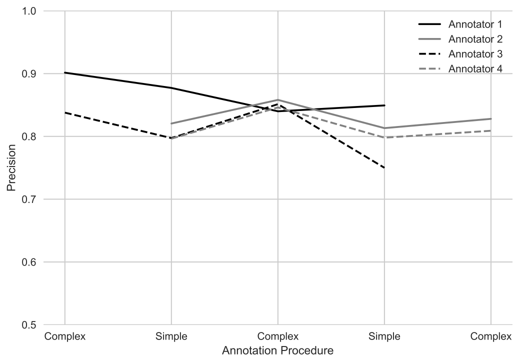

Figure 6: Precision as a function of the simple versus complex annotation procedure. The y-axis of the figure displays average precision across all utterance-code pairs.

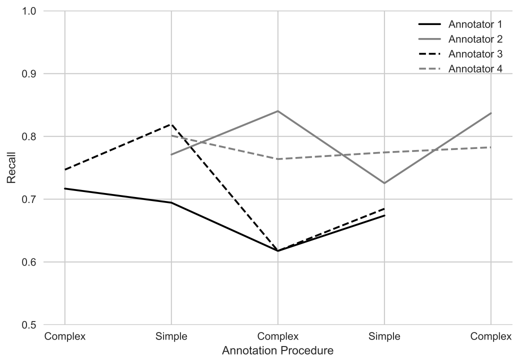

Figure 7: Recall as a function of the simple versus complex annotation procedure. The y-axis of the figure displays average recall across all utterance-code pairs.

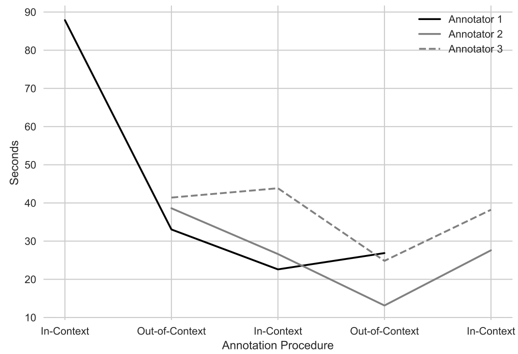

Figure 8: Average total annotation time per utterance as a function of the context provided to annotators. In both procedures, annotators code for eight codes at once, but while the "In Context" procedure shows all utterances in order, the "Out of Context" procedure present coach utterances in randomized order. The y-axis of the figure displays the total average annotation time per utterance.

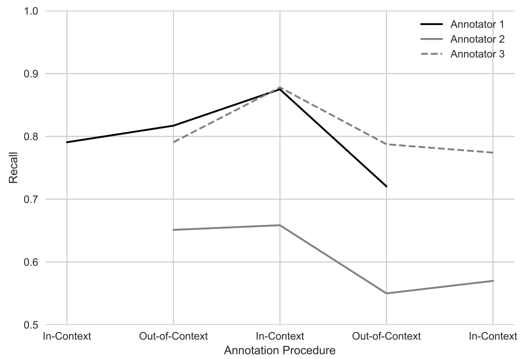

Figure 9: Recall as a function of the context provided to annotators. In both procedures, annotators code for eight codes at once, but while the "In Context" procedure shows all utterances in order, the "Out of Context" procedure present coach utterances in randomized order. The y-axis of the figure displays average accuracy across all utterance-code pairs.

List of Tables

Table 1: Coding scheme.

Strategy	Definition	Example
Positive Evaluation	Positive judgement about a teacher's skills or practice	*You are so kind and engaging with him!*
Observation	Specific information about the students or teacher based on the coach's observation	*You tend to ask kids to raise their hands a lot.*
Suggestion	Explicit proposal that the teachers can or should make to their instruction	*One thing you could do is to try to avoid a negative tone of voice.*
Instruction	Information that helps a teacher understand the importance or purpose of an instructional strategy	*Being more specific with your redirections ensures that your students understand your expectations and can follow them.*
Demonstration	A specific demonstration of how to implement and instructional strategy	*A calm tone of voice would sound like,"Ethan, please be quie".*
Anticipation	A question that prompts the teacher to elaborate on the consequences of an instructional strategy	*What do you think would happen if you asked students to raise their hands?*
Practice	Dialogue where the coach facilitates a role-play activity	*We're going to practice. I'll pretend to be a student and I want you to redirect me.*

Table 2: Study design and annotation procedure assignments.

Annotator	Week 1	Week 2	Week 3	Week 4
1	Complex	Simple	Complex	Simple
2	Simple	Complex	Simple	Complex
3	Complex	Simple	Complex	Simple
4	Simple	Complex	Simple	Complex

Table 3: Impact of the simple annotation procedure on accuracy, precision, and recall.

	Accuracy ($M = 0.95$)	Precision ($M = 0.75$)	Recall ($M = 0.83$)
Annotator 1*Simple Scheme	-0.003	-0.003	0.041
	(0.008)	(0.037)	(0.041)
Annotator 2*Simple Scheme	-0.011	-0.035	-0.119**
	(0.008)	(0.036)	(0.036)
Annotator 3*Simple Scheme	-0.006	-0.052	0.103**
	(0.009)	(0.04)	(0.04)
Annotator 4*Simple Scheme	-0.001	-0.043	-0.014
	(0.008)	(0.038)	(0.04)
Average Impact Across Annotators	-0.005*	-0.035**	0.003
	(0.003)	(0.012)	(0.015)

Note. $N = 508$. The first four rows of the table represent the impact of the simple annotation procedure on each of the four annotators' accuracy, precision, and recall, estimated using Equation 1. The final row represents the average impact of the simple annotation procedure across all four annotators, estimated using Equation 2. ***$p < 0.001$, **$p < 0.01$, *$p < 0.05$.

Table 4: Impact of the lack of context (when annotating for all codes at once) on accuracy, precision, and recall.

	Accuracy ($M = 0.95$)	Precision ($M = 0.74$)	Recall ($M = 0.85$)
Annotator 1*No Context Scheme	0.005	-0.029	-0.03
	(0.009)	(0.044)	(0.052)
Annotator 2*No Context	0.005	-0.029	-0.03
	(0.009)	(0.044)	(0.052)
Annotator 3*No Context	0.005	-0.009	-0.054
	(0.009)	(0.048)	(0.047)
Average Impact Across Annotators	-0.002	0.007	-0.043*
	(0.003)	(0.016)	(0.022)

Note. $N = 360$. The first three rows of the table represent the impact of the lack of context, when coding for eight codes at once, on accuracy, precision, and recall for each of three annotators, estimated using Equation 1. The final row represents the average impact of lack of context across all three annotators, estimated using Equation 2. ***$p < 0.001$, **$p < 0.01$, *$p < 0.05$.