Automated Content Misclassification Causes Bias in Regression. Can We Fix It? Yes We Can!

Abstract

Automated classifiers have become widely popular measurement devices in communication science. These classifiers, often built via supervised machine learning (SML), can categorize large, statistically powerful samples of data ranging from text to images and video. Automated classifiers make errors and therefore regression analyses using them invariably produce biased inferences—unless analyses account for these errors. As we show in a systematic literature review of SML applications, communication scholars rarely acknowledge this important problem of "ignoring misclassification in automated content analysis". In principle, existing statistical methods that use "gold standard" validation data, such as that created by human annotators, can account for misclassification and produce correct statistical results. We introduce and test such methods, including a new method we design and implement in the R package `misclassificationmodels`, via Monte-Carlo simulations designed to reveal each method's limitations. Based on these results, we provide recommendations for addressing misclassification errors via statistical correction methods. In sum, automated classifiers, even those below common accuracy standards, can be useful for measurement with careful study design and appropriate correction methods.

Keywords: Content Analysis; Machine Learning; Classification Error; Attenuation Bias; Simulation; Computational Methods; Big Data; AI;

Automated Content Misclassification Causes Bias in Regression. Can We Fix It? Yes We Can!

Automated classifiers (ACs) based on supervised machine learning (SML) have rapidly gained popularity as part of the *automated content analysis* toolkit in communication science (Baden et al., 2022). With these measurement devices, researchers can categorize large samples of text, images, video or other types of data into predefined categories (Scharkow, 2017). In communication science, studies for instance use ACs to automatically classify topics (Vermeer et al., 2020) or frames (Opperhuizen et al., 2019) in news articles or social media posts.

However, there is increasing concern about the validity of automated content analysis (Baden et al., 2022; Grimmer & Stewart, 2013). ACs make *misclassifications* which cause bias in statistical inferences—unless correctly modeled (Fong & Tyler, 2021; Scharkow & Bachl, 2017). Research areas where ACs have the greatest potential—e.g., content moderation, social media bots, affective polarization, or radicalization—are haunted by the specter of methodological questions related to misclassification (Baden et al., 2022; Rauchfleisch & Kaiser, 2020): How accurate must an AC be to usefully measure a variable? When—if ever—should an AC built for one context be used in another (González-Bailón & Paltoglou, 2015; Hede et al., 2021)? How do biases that an AC learns from training data affect findings of downstream analyses (Millimet & Parmeter, 2022)? Knowing that high classification accuracy limits the risks of misleading inference, careful researchers might use only those ACs having excellent predictive performance. Yet, important social scientific concepts such as as sentiment (van Atteveldt et al., 2021) civility (Hede et al., 2021), and institutional frameworks (Rice et al., 2021) can be challenging to classify with high performance.

In a systematic literature review of $N = 48$ studies employing SML-based text classification to study substantial empirical questions, we show that the problem of *ignoring misclassification* is widespread. Our review demonstrates a troubling lack of attention to the threats ACs introduce—and virtually no mitigation of such threats. In the current state of affairs, ACs are unlikely to be useful for studying nuanced concepts. Researchers will either draw misleading conclusions from inaccurate ACs or avoid ACs in favor of costly methods such as manually coding large samples (van Atteveldt et al., 2021).

In this study, we therefore *discuss and test different statistical methods for addressing misclassification* with the goal of rescuing ACs from this dismal state (Buonaccorsi, 2010; Carroll et al., 2006; Yi et al., 2021). These methods include Fong and Tyler (2021)'s generalized method of moments (GMM) calibration method, Zhang (2021)'s pseudo-likelihood models, and Blackwell et al. (2012)'s application of imputation methods. In addition, we develop our own specialized implementation of a general likelihood modeling framework drawn from the statistical literature on measurement error (Carroll et al., 2006), which we implement via the experimental R package `misclassificationmodels`.

We test these methods via Monte Carlo simulations of four prototypical situations representative of those identified by our systematic review: Using ACs to measure either (1) a dependent or (2) an independent variable where the classifier makes misclassifications that are either (a) easy to correct (e.g., nondifferential error) or (b) more difficult (e.g., differential error). The more difficult cases are important: For example, errors made by classifiers affect moderation behavior in online communities (*1 citation removed for masked review*). Studies using classifiers to study moderation may therefore be prone to differential error, which can cause misleading statistics even when classification accuracy is high. While each of the methods from prior work improves upon common practice, none are effective in all scenarios. When correctly applied, our likelihood modeling is the only correction method recovering the true parameters in all scenarios.

We are optimistic about the potential of ACs for communication science and beyond. According to our simulations, even classifiers with low predictive performance can be useful as long as human validation data is adequate. Researchers can even fruitfully use biased ACs—as long as the bias can be modeled. However, we also show that is not enough to "validate" ACs and make misclassification rates transparent: Researchers also have to statistically correct for measurement errors. The required assumptions for error correction methods are no more difficult than those already commonly adopted in traditional content analyses—and much more reasonable than the current default assumption that ACs are good enough.

In sum, this paper makes a methodological contribution by introducing the often-ignored problem of "ignoring misclassification in automated content analysis", by testing approaches to address this problem via Monte Carlo simulations, and by introducing a new method for error correction. This method can succeed where others fail, is easily applied by experienced regression modelers, and is straightforward to extend.

Misclassification in Automated Content Analysis: Reviewing Reporting and Error Correction Practices

Content analysis focuses on "*making replicable and valid inferences from texts (or other meaningful matter) to the contexts of their use*" (Krippendorff, 2018, p. 24, emphasis in original). Automated content analysis, where computers are used as measurement devices, has gained traction in communication science (Baden et al., 2022; Jünger et al., 2022) (*1 citation removed for masked review*). One common automated content analysis method is supervised machine learning (SML) (Scharkow, 2017).[1] In essence, the procedure is to train an algorithm—e.g., a naïve Bayes classifier, decision tree, or artificial neural network—on manually coded material as the training set. The trained classifier is then used to predict categories in new, as of yet unseen data. Automatic classifiers enable researchers to inexpensively measure categorical variables in large data sets of digitized media. This

[1] Automated content analysis includes a range of other methods both for assigning content to predefined categories (e.g., dictionaries) and for assigning content to unknown categories (e.g., topic modeling) (Grimmer & Stewart, 2013). Here, we focus on SML-based ACs. However, our arguments extend to other deductive approaches introducing misclassifications such as dictionary-based classification.

promises to be useful for study designs requiring large samples such as to infer effect sizes smaller than would be possible using a sample size that humans could feasibly classify.

But are scholars aware that misclassification by ACs poses threats to the validity of downstream analyses? Although such issues in the context of manual content analysis have attracted much debate (Bachl & Scharkow, 2017), this is less true for misclassification by newly popular automatic classifiers. To understand how social scientists, including communication scholars, use SML-based classifiers to construct variables and engage with the problem of misclassification, we conducted a systematic literature review (see Appendix A in our Supplement for details[2]). Our review builds on studies identified by recent reviews on automated content analysis, including SML (Baden et al., 2022; Hase et al., 2022; Jünger et al., 2022; Song et al., 2020). Our goal in our review is not to comprehensively review all SML studies but to provide a picture of common practices, with an eye toward awareness of misclassification and its statistical implications.

We identified a total of 48 empirical studies published between 2013 and 2021—more than half of which were published in communication journals—which employed SML-based text classification to create 146 variables. Studies used SML-based text classification to perform tasks stuch as identifying frames (Opperhuizen et al., 2019) or topics (Vermeer et al., 2020). They often employed SML-based ACs to create dichotomous (50%) or other categorical (22.9%) variables[3]. Although 89.6% of empirical studies used SML-based ACs to report descriptive statistics, many also employed automated classification for downstream statistical analyses by using ACs as dependent (43.8%) and independent (39.6%) variables in multivariate models. These multivariate analyses tend to be reported in higher-status journals compared to papers only reporting proportions.

Given the rising popularity of SML-based text classification, our review indicates a

[2] Anonymized link for review: https://osf.io/pyqf8/?view_only=c80e7b76d94645bd9543f04c2a95a87e

[3] Metric variables were also created in 35.4% of studies, mostly via the non-parametric method by Hopkins and King 2010 estimating proportions instead of classifying documents, something we do not focus on.

worrying *lack of transparency when reporting SML-based text classification*, similar to that reported in previous studies (Reiss et al., 2022): A large share of studies do not report important methodological decisions related to the sampling and sizes of training and test sets or to intercoder reliability (see Appendix A). This lack of transparency concerning model validation not only limits the degree to which researchers can evaluate studies, but also makes replicating such analyses to correct for misclassification nearly impossible. Most importantly, our review finds that *studies almost never reflected upon or corrected for misclassification in their automated content analyses.* According to our review, only 18.8% of studies discussed in any way the possibility that an AC misclassified texts. Only a single article reported using error correction methods.

The Neglected Problem of Misclassification

Misclassification is a long-standing concern in the content analysis literature which has extensively studied difficulties in human-labeling through the framework of intercoder reliability (Krippendorff, 2004). The increasing use of metrics such as Krippendorf's α demonstrates transparency efforts in reporting imperfect manual annotations (Lovejoy et al., 2014). Moreover, Bachl and Scharkow (2017) introduced methods for correcting proportion estimates using data from multiple independent human coders. Despite this awareness of threats posed by manual misclassification, our review demonstrates that misclassification by ACs is often downplayed.

Unless an AC is perfect, it makes errors. This misclassification causes bias in statistical inference, particularly in the context of regression models (Carroll et al., 2006; Scharkow & Bachl, 2017). A large dataset does not reduce such inferential bias (Carroll et al., 2006; van Smeden et al., 2020). It is often believed—incorrectly—that misclassification causes only conservative bias (i.e., bias towards 0) because this is true in the simplest cases of least squares regression—when measurement error in the only covariate is classical or when measurement error in the outcome is unbiased (Carroll et al., 2006; Loken & Gelman, 2017; van Smeden et al., 2020).[4] As a result, researchers interested in a hypothesis of a statistically significant relationship may not consider misclassification an important threat to validity (Loken & Gelman, 2017). However, there are at least two compelling reasons that misclassification is a serious concern.

Problem I: Misclassification can cause anti-conservative bias

First, the inferential bias that misclassification causes is not necessarily conservative (Carroll et al., 2006; Loken & Gelman, 2017; van Smeden et al., 2020). In logistic regression or other nonlinear models, random measurement error can cause bias away from 0. Moreover differential measurement error (i.e., error not conditionally independent of the outcome given the other covariates) can bias inference in any direction and lead to wildly misleading conclusions. Researchers can check the assumption of nondifferential measurement error via graphical and statistical conditional independence tests (Carroll et al., 2006; Fong & Tyler, 2021). For example, Fong and Tyler (2021) suggest using Sargan's J-test of the null hypothesis that the product of the AC's predictions and regression residuals have an expected value of 0.

Users of ACs should be especially conscious of differential measurement error due to the nonlinear behavior of many ACs (Breiman, 2001). For instance, ACs designed in one context and applied in another are likely to cause differential measurement error. The Perspective API used to classify toxic content, for example, was developed for social media comments, but performs much worse when applied to news data (Hede et al., 2021). Differential measurement error is also likely to arise when an AC used for measurement

[4] Measurement error is *classical* when $W = X + \xi$ because the variance of an AC's predictions is greater than the variance of the true value (Carroll et al., 2006). If nondifferential measurement error is not classical then it is called Berkson, and we would write $X = W + \xi$ instead of $W = X + \xi$. In general, Berkson measurement error is easier to deal with than classical error. It is hard to imagine how a AC would have Berkson errors (the predictions would have to have lower variance than the training data), so, following prior work, we do not consider Berkson errors (Fong & Tyler, 2021; Zhang, 2021).

shapes behavior in the sociotechnical system under study. For example, the Perspective API is used for moderation in many forums (Hede et al., 2021). Therefore, its predictions may have causal effects on outcomes related to moderation which cause differential error in regression models using these ACs as covariates.

Problem II: Systematic Biases in Specific Research Areas

The second reason that misclassification is a concern is that it may systematically contaminate the literature in a research area. If certain ACs become standard measurement devices within a research area, such as the LIWC dictionary to measure sentiment (Boukes et al., 2020), Google's Perspective API used to measure toxicity (Hosseini et al., 2017) or Botometer used to classify social media bots (see, for a critical discussion Rauchfleisch & Kaiser, 2020), such research areas may become confused by systematic biases. For example, Scharkow and Bachl (2017) argue that media's "minimal effects" on political opinions and behavior may be an artifact of how many study designs in this area have common sources of measurement error that created systematic bias towards 0. Conversely, if researchers selectively report statistically significant hypothesis tests, measurement error can introduce an upward bias in the magnitude of reported effect sizes and contribute to a replication crisis (Loken & Gelman, 2017).

Is Transparancy about Misclassification Enough?

Commonly recommend practices in automated content analyses address the threats of misclassification through *transparency* in the form of reporting metrics such as precision, recall, F1 and AUC scores computed using human-classified validation data (Grimmer & Stewart, 2013). These metrics are intended to promote confidence in inferences resulting from the use of ACs by demonstrating high predictiveness. However, our literature review indicates that they are not always included in reporting, at least when it comes to SML-based text classifications.

Moreover, high predictiveness according to these metrics may be less protective from measurement error than it seems. Algorithms and models for building effective automated classifiers were developed in the culture of algorithmic modeling associated with fields like computer science and management (Breiman, 2001). As a paradigm, SML takes the opposite position on the bias-variance tradeoff from conventional statistics. Its methods achieve high predictiveness by throwing unbiased inference to the wind and pursuing prediction at all costs (Breiman, 2001). On their own, predictiveness metrics provide no guarantees about the accuracy of downstream statistical inferences.

In fact, steps made in the interest of predictiveness may increase inferential bias. As a growing body of scholarship critical of the hasty adoption of SML in criminal justice, healthcare, content moderation, and employment has demonstrated, machine learning models boasting high performance often have biases. These result from the use of non-representative training datasets and spurious correlations that neither reflect causal mechanisms nor generalize in different (sub)populations (Bender et al., 2021). For example, Hede et al. (2021) show that, when applied to news datasets, the Perspecitve API overestimates incivility in topics such as racial identity, violence and sex. These automatic classifications will likely introduce differential measurement error to a regression model of an outcome related to such topics. If ACs used in communication science also have such biases, these biases may flow downstream, by way of differential or systematic measurement error, into statistical inferences.

The good news is that human-classified validation data can do more than benchmark predictive performance to increase transparency about measurement errors. With an appropriate model, validation data can effectively correct biases in statistical inferences.

Correcting for Misclassification

Statisticians have extensively studied problems that measurement errors can cause for statistical inferences and proposed statistical methods to correct them (see Carroll et al., 2006; Fuller, 1987). We therefore narrow our focus to methods that are particularly appropriate to dealing with misclassifications by ACs: Fong and Tyler (2021)'s GMM calibration method, Zhang (2021)'s pseudo-likelihood model, and approaches that promise greater generality—multiple imputation, (Blackwell et al., 2012) and likelihood modeling (Carroll et al., 2006).

In the interest of clarity, we introduce some notation in this section. Say X is the covariate that is automatically classified, and X^* is a sample of validation data. The automatic classifications are W, Z is a second covariate, and Y is the outcome. To illustrate, consider an idealized example study from social media research: whether someone breaks a rule on a social media site and how long it takes for them to be banned. This study might analyze the regression model $Y = B_0 + B_1X + B_2Z + \varepsilon$ where Y is the (log-scaled) time until an account is banned, X is whether the account broke a rule, and Z is a covariate related to the account's reputation, such as the number of posts. Humans can observe whether an account breaks a rule, but human classifications are expensive and only available in a relatively small sample X^*. In contrast, an SML model can make automatic classifications W for the entire dataset. But how do we correct for errors introduced by such ACs?

Regression calibration uses observable variables, including the automatic classifications W and other variables measured without error Z, to approximate the true value of a covariate X (Carroll et al., 2006). Fong and Tyler (2021) propose a regression calibration procedure designed for supervised machine learning that we refer to as *GMM calibration* or abbreviate as GMM.[5] For their calibration model, Fong and Tyler (2021) use 2-stage least squares (2SLS), regressing observable covariates Z and AC predictions W onto the validation data and then use the resulting model to approximate the covariate $\hat{X}$. Next, Fong and Tyler (2021) use the generalized method of moments (gmm) to combine the estimate based on the approximated covariate $\hat{X}$ and the estimate using the validation

[5] Fong and Tyler (2021) describe their method within an instrumental variable framework, but it is equivalent to regression calibration and regression calibration is the standard term in measurement error literature.

data X^*. This method makes efficient use of validation data and provides an asymptotic theory for deriving confidence intervals. The GMM method's assumptions do not include strong assumptions about the distribution of the outcome Y, but are still violated by differential error (Fong & Tyler, 2021). GMM, like other regression calibration techniques, is not designed to correct for misclassification in the outcome.

Multiple imputation (MI) treats measurement error as a missing data problem because the true value of X is observed in the validation data X^* and missing otherwise (Blackwell et al., 2012). For example, the regression calibration step in Fong and Tyler (2021)'s GMM method uses least squares regression to impute unobserved values of the covariate X. Indeed, Carroll et al. (2006) describe regression calibration when validation data are available as "simply a poor person's imputation methodology" (pp. 70). Like regression calibration, multiple imputation uses a model to infer likely values of possibly misclassified variables. The difference is that multiple imputation samples several (hence *multiple* imputation) entire datasets filling in the missing data from the predictive probability distribution of the covariate X conditional on the other variables $\{X, Y, Z\}$, then runs a statistical analysis on each of these sampled datasets and pools the results of each of these analyses (Blackwell et al., 2012). Note that Y is included among the imputing variables, giving the MI approach the potential to address differential error. Blackwell et al. (2012) claim that their MI method works with differential measurement error (so long as the bias in the measurement error can be modeled) and when measurement error is in the outcome or in a covariate.

Maximum likelihood methods (MLE) can effectively deal with measurement error in ACs by maximizing a likelihood that correctly specifies an *error model* of the probability of the automatic classifications conditional on the true value and the outcome (Carroll et al., 2006). In contrast to the GMM and the MI approach, which predict values of the mismeasured variable, the MLE method accounts for all possible values of the variable by "integrating them out" of the likelihood. "Integrating out" means adding both possible values of a binary variable to the likelihood, weighted by the likelihood of the error model. MLE methods have two advantages in the context of ACs. First, they are quite general and can be applied to any model with a convex likelihood including generalized linear models (GLMs) and generalized additive models (GAMs). Second, assuming the model is correctly specified, MLE estimators are fully consistent whereas regression calibration estimators are only approximately consistent (Carroll et al., 2006). Practically, this means that MLE methods can have greater statistical efficiency and require less validation data to make precise estimates.

The MLE approach is conceptually different from the GMM one. The GMM approach first imputes likely values and then runs the main analysis on imputed values. By contrast, MLE approaches estimate—all in one step—the main analysis using the full dataset and the error model estimated using only the validation data (Carroll et al., 2006). The MLE approach is applicable both when the automatically classified variable is a covariate and when it is the outcome.

"Pseudo-likelihood" methods (PL)—even if not always explicitly labeled this way—are another approach. Zhang (2021) proposes a method that approximates the error model using quantities from the AC's confusion matrix—the positive and negative predictive values in the case of a mismeasured covariate and the AC's false positive and false negative rates in the case of a mismeasured outcome. Because quantities from the confusion matrix are neither data nor model parameters, Zhang (2021)'s method is technically a "pseudo-likelihood" method. A clear benefit of this idea is that it only requires summary quantities derived from validation data. It can thus be applied when validation data are unavailable. We will discuss likelihood methods in greater depth in the presentation of our MLE framework below.

Statisticians have studied other methods for correcting measurement error that we do not test in our simulations including simulation extrapolation, Bayesian estimation, and score function methods. As we argue in Appendix B of our Supplement, these approaches are not advantageous for correcting misclassification when validation data is available.

Proposing a Likelihood Modeling Approach to Correct Misclassification

We now elaborate on our likelihood modeling approach by applying Carroll et al. (2006)'s presentation of the general statistical theory of likelihood modeling for measurement error correction to the context of binary classification when validation data is available. The idea is to use an *error model* of the conditional probability of the automatic classifications given the true classifications and other variables on which automatic classifications depend. In other words, the error model estimates the conditional probability mass function of the automatic classifications.

Including the error model in the likelihood effectively accounts for uncertainty of the true classifications and, assuming the error model gives consistent estimates of the conditional probability of the automatic classifications given the true values, is sufficient to obtain consistent estimates using MLE (Carroll et al., 2006). The MLE approach is particularly well-suited to misclassification by ACs because it can be quite straightforward to fit the error model when the mismeasured variable is discrete.

When an Automatic Classifier Predicts a Covariate

Say we want to fit the linear regression model $Y = B_0 + B_1X + B_2Z + \varepsilon$ and an AC makes classifications W that predict the discrete covariate X—for instance, whether a message by a social media account broke a rule according to an AC, to then explain the time until the account is banned. Maximizing $(\mathcal{L}(\Theta|Y, W))$, the likelihood of parameters Θ given data W and Y, can jointly fit the regression model of Y having parameters $\Theta_Y = \{B_0, B_1, B_2\}$ and an error model of W because $P(Y, W|\theta)$, the joint probability of Y and W, can be factored into the product of three terms: $P(Y|\Theta_Y)$, the regression model we want to fit, $P(W|X, Y)$, the error model, and $P(X|Z)$, a model for the probability of X. Therefore, calculating these three conditional probabilities is sufficient to calculate the joint probability of the outcome and automatic classifications and obtain a consistent estimate despite misclassification.

For instance, we can assume that the probability of W follows a logistic regression model of Y, X and Z and that the probability of X follows a logistic regression model of Z. In this case, the likelihood model below is sufficient to consistently estimate the parameters $\Theta = \{\Theta_Y, \Theta_W, \Theta_X\} = \{\{B_0, B_1, B_2\}, \{\alpha_0, \alpha_1, \alpha_2\}, \{\gamma_0, \gamma_1\}\}$.

$$\mathcal{L}(\Theta|Y,W) = \prod_{i=0}^{N} \sum_{x} P(Y_i|X_i, Z_i, \Theta_Y) P(W_i|X_i, Y_i, Z_i, \Theta_W) P(X_i|Z_i, \Theta_X) \tag{1}$$

$$P(Y_i|X_i, Z_i, \Theta_Y) = \phi(B_0 + B_1 X_i + B_2 Z_i) \tag{2}$$

$$P(W_i|X_i, Y_i, Z_i, \Theta_W) = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 Y_i + \alpha_2 X_i)}} \tag{3}$$

$$P(X_i|Z_i, \Theta_X) = \frac{1}{1 + e^{-(\gamma_0 + \gamma_1 Z_i)}} \tag{4}$$

ϕ is the normal probability distribution function. Note that Equation 1 models differential error taking the form of a linear relationship between W and Y. When error is nondifferential, the dependence between W and Y can be removed from Equations 1 and 3.

Calculating the three conditional probabilities in practice requires specifying models on which validity of the method depends. This framework is very general and a wide range of probability models, such as generalized additive models (GAMs) or Gaussian process classification, may be used to estimate $P(W|X,Y)$ and $P(X|Z)$ (Williams & Barber, 1998). For simplicity, we proceed with a focus on linear regression for the probability of Y and logistic regression for the probability of W and the probability of X.

When an Automatic Classifier Predicts the Outcome

We now turn to the case when an AC makes classifications W that predict the discrete-valued outcome Y—for example to use an automatically classifier predicting whether social media users break rules to test hypotheses about why they do so. This case is simpler than the case above where an automatic classifier is used to measure a covariate X because there is no need to specify a model for the probability of X.

If we assume that the probability of Y follows a logistic regression model of X and Z, and allow W to be biased and directly depend on X and Z, then maximizing the following likelihood is sufficient to consistently estimate the parameters $\Theta = \{\Theta_Y, \Theta_W\} = \{\{B_0, B_1, B_2\}, \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}\}$.

$$\mathcal{L}(\Theta|Y,W) = \prod_{i=0}^{N} \sum_{x} P(Y_i|X_i, Z_i, \Theta_Y) P(W_i|X_i, Z_i, Y_i, \Theta_W) \tag{5}$$

$$P(Y_i|X_i, Z_i, \Theta_Y) = \frac{1}{1 + e^{-(B_0 + B_1 X_i + B_2 Z_i)}} \tag{6}$$

$$P(W_i|Y_i, X_i, Z_i, \Theta_W) = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 Y_i + \alpha_2 X_i + \alpha_3 Z_i)}} \tag{7}$$

If the AC's errors are conditionally independent of X and Z given the model for W then the dependence of W on X and Z can be omitted from equations 5 and 7. Additional details are available in Appendix C of the Supplement.

Simulation Design

In this section, we present four Monte Carlo simulations (*Simulations 1a*, *1b*, *2a*, and *2b*) to evaluate existing methods (GMM, MI, PL) as well as our approach (MLE) for correcting statistical inference when a variable is measured by an error-prone AC. We first describe the set-up of our Monte Carlo simulations before delving into the four prototypical scenarios we identified via our literature review and therefore simulated.

Parameters of the Monte Carlo simulations

Monte Carlo simulations are a common tool for evaluating statistical methods, including (automated) content analysis (e.g. Bachl & Scharkow, 2017; Fong & Tyler, 2021; Geiß, 2021; Song et al., 2020; Zhang, 2021). A Monte Carlo simulation defines a model of study design in terms of a data generating process from which datasets are repeatedly sampled. Running an analysis on each sampled dataset provides an empirical distribution of the results the analysis would obtain over study replications. The methods affords exploration of finite-sample performance, robustness to assumption violations, comparison across several methods, and ease of interpretability (Mooney, 1997).

For each prototypical scenario, we ran up to six analyses. Four of these test error correction methods: *GMM calibration* (GMM) (Fong & Tyler, 2021), *multiple imputation*

(MI) (Blackwell et al., 2012), *Zhang's pseudo-likelihood model* (PL) (Zhang, 2021), and our *likelihood modeling* (MLE) approach. GMM is not designed for the case when an automatically classified variable is the outcome, so we omit this method in *Simulations 2a* and *2b*. We compare error correction methods to two other approaches: the *feasible* estimator in which researchers abstain from using ACs by using only perfectly accurate manually annotated validation data (i.e., cases where manual coders agree on codes) and the *naïve* estimator, representative of common practice, where researchers use AC- based classifications W as a stand-ins for X.

We repeat each simulation with different amounts of automatically classified data (ranging from 1000 to 10000 observations) and human labeled data (ranging from 100 to 400 observations).

$$Y = B_0^* + B_1^* W + B_2^* Z + \varepsilon^* = B_0^* + B_1^*(X + \xi) + B_2^* Z \tag{8}$$

We evaluate each analytical approach in terms of *consistency*, whether the estimates of parameters $\hat{B_X}$ and $\hat{B_Z}$ have expected values nearly equal to the true values B_X and B_Z; *efficiency*, how precisely the parameters are estimated and how precision improves with additional automatically classified or human labeled data; and *uncertainty quantification*, how well the 95% confidence intervals provided by each method approximate the confidence interval of parameter estimates across Monte Carlo simulations.

We use the `predictionError` R package (Fong & Tyler, 2021) for the GMM method, the `Amelia` R package for the MI approach, and the `optim()` R function for implementing Zhang (2021)'s PL approach and our approach.

Four Prototypical Scenarios

We simulate regression models with two covariates (X and Z). This sufficiently constrains our study's scope but is general enough to be applied in a wide range of research studies. Whether the methods we evaluate below are effective or not depends on the conditional dependence structure among the covariates, the outcome Y, and the model predictions W. This structure determines whether covariate measurement error is differential and whether outcome measurement error is systematic (Carroll et al., 2006). We illustrate our simulated scenarios using Bayesian networks to represent the conditional dependence structure of the variables in Figure 1 (Pearl, 1986).

We first simulate two cases when an AC is used to measure a covariate with and without differential error. Then, we simulate two cases where an AC is used to measure the outcome either making errors that are correlated with predictors or not.

Measurement Error in a Covariate (***Simulations 1a*** and ***1b***)

We consider studies with a goal of testing a hypotheses about the coefficients B_1 and B_2 in the least squares regression (Model 9).

$$Y = B_0 + B_1X + B_2Z + \varepsilon \tag{9}$$

In this example, Y is continuous variable, X is a binary variable measured with an AC, and Z is a normally distributed variable with mean 0 and standard deviation 0.5 measured without error. For example, Y could be the time until an account on an online forum is banned, X if a message breaks one of the forum's rules, and Z the account's reputation score. X and Z are negatively correlated because high-reputation accounts may be less likely to break rules.

Say that human content coders can observe X perfectly, but each observation is so expensive that observing X for a large sample is infeasible. To scale up content analysis, a SML-based AC makes predictions W of X—for instance predicting if any of the messages from that social media user break the rules. Both scenarios have a normally distributed outcome Y and two binary-valued covariates X and Z, which are balanced ($P(X) = P(Z) = 0.5$) and correlated (Pearson's $\rho = -0.12$). Simulating balanced covariates serves simplicity so that accuracy is adequate to quantify the predictive performance of our simulated classifier. Simulating correlated covariates is helpful to study how misclassification in one variable affects parameter inference in other covariates. To represent a research study design where automated classification is needed to obtain sufficient statistical power, Z and X can explain only 10% of variance in Y.

In *Simulation 1a*, visualized in Figure 1a, we simulate an AC with 72% accuracy to reflect a situation where X may be difficult to predict, but an automated classifier, represented as a logistic regression model having linear predictor W^*, provides a useful signal. The *naïve estimator* has classical and nondifferential measurement error because $W = X + \xi$ because ξ is normally distributed with mean 0 and ξ is conditionally independent of Y given X and Z ($P(\xi|Y, X, Z) = P(\xi|X, Z)$).

In *Simulation 1b* visualized in Figure 1b, the AC's predictions directly depend on the outcome Y, so we can test error correction methods in the presence of differential error. We create this dependence by simulating an AC with 74% accuracy that makes predictions W that are negatively correlated with the residuals of the linear regression of X and Z on Y (Pearson's $\rho = -0.17$). As a result, this AC makes fewer false-positives and more false-negatives at greater levels of Y. Although the false-negative rate of the AC is 25% overall, when $Y <= 0$ the false-negative rate is only 14%, but when $Y >= 0$ it rises to 32%.

These simulations are prototypical of an AC that influences behavior in a system under study such as if community moderators use ACs to identify rule-breakers and correct their behavior. False negatives may cause delays in moderation increasing Y (time-until-ban), while false-positives could draw moderator scrutiny and cause them to issue speedy bans. This mechanism is not mediated by observable variables such as reputation (Z) or the true rule-breaking (X). Therefore, Model 8 has differential error.

Measurement Error in the Outcome (Simulation 2a and 2b)

We then simulate using an AC to measure the dependent variable Y, a binary covariate X, and a continuous covariate Z. For example, Y describes whether a message is rule-breaking, X whether the user leaving the message has been warned by moderators, and Z a reputation score. The goal is to estimate B_1 and B_2 in the following logistic regression model:

$$P(y) = \frac{1}{1 + e^{-(B_0 + B_1 x + B_2 z)}} \tag{10}$$

As was true for X in *Simulation 1*, human coders can observe Y, but at considerable expense, and an AC makes predictions $W = Y + \xi$.

Simulation 2a (visualized in Figure 1c) and *Simulation 2b* (visualized in Figure 1d) implement these scenarios. Here, X and Z are balanced $P(X) = P(Z) = 0.5$ and correlated. (Pearson's $\rho = -0.12$). As in *Simulation 1* we simulate scenarios where an AC is of practical use to estimate subtle relationships. In *Simulation 1* we chose the variance of the normally distributed outcome given our chosen coefficients B_X and B_Z, but this is not appropriate for *Simulation 2*'s logistic regression so we choose, somewhat arbitrarily, $B_X = 0.7$ and $B_Z = -0.7$.

Again, we simulate ACs with moderate predictive performance. The AC in *Simulation 2a* is 72% accurate and the AC in *Simulation 2b* is 78% accurate. In *Simulation 2a*, the predictions W are unbiased because classification errors ξ have mean 0 and are independent of covariates X and Z. However, in *Simulation 2b* the predictions are biased because their errors ξ are correlated with Z (Pearson's $\rho = -0.16$). One way such a correlation might obtain in our example of online moderation is if community members are adept at skirting the rules without violating them. Such members are both likely to be warned by moderators and also to leave messages misclassified as rule-breaking.

Simulation Results

We visualize the consistency, efficiency, and the accuracy of uncertainty quantification of each method in each prototypical scenario. For example, Figure 2 visualizes results for *Simulation 1a.* Its subplots each show a simulation with a given total sample size (No. observations) and validation sample size (No. validation data).

To understand how each plot visualizes the consistency of estimators, see for instance the leftmost column in the bottom-left subplot illustrating performance of the naïve estimator using AC classifications W to stand in for the true variable X. The center of the black circle locates the expected value of the point estimate over our 500 simulations. For the naïve estimator in Figure 2, the circle is far below the dashed line which shows the true value of B_X, indicating that misclassification causes a dramatic bias toward 0 and that the estimator is inconsistent.

To assess efficiency, we mark the region in which point estimate falls in 95% of the simulations with black lines. These black lines in the bottom-left subplot of Figure 2 for example show that the feasible estimator, which uses only perfectly accurate validation data, is consistent but less precise than the estimates from correction methods that use both automatic classifications and human-labeled data.

The accuracy of the method's uncertainty quantification can be seen by comparing the gray lines, which show for each method the expected value of its approximate 95% confidence intervals over the 500 simulations for each method, to the neighboring black lines. The *PL* column in the bottom-left subplot of Figure 2 shows that the method's 95% confidence interval is biased towards 0 when the number of human labels is low. This result is expected because the method does not account for uncertainty in misclassification probabilities estimated using the sample of true classifications. Now that we have explained how to interpret our plots, we will unpack them for each simulated scenario.

Simulation 1a: When Misclassifications Are Independent of the Outcome

As visualized in Figure 2, the naïve estimator is severely biased in its estimation of B_X in *Simulation 1a.* Fortunately, error correction methods including our MLE method as well as the GMM and MI approach produce consistent estimates and acceptably accurate confidence intervals. Notably, the PL method is inconsistent and considerable bias remains when the number of human classifications is much less than the total number of observations. The most likely source of this inconsistency is that $P(X = x)$ is missing from the pseudo-likelihood as can be seen by comparing Equation C4 in our Supplement to Equations 24-28 from Zhang (2021). The bottom row of Figure 2 shows that the precision of MLE and GMM estimates increase in larger datasets. However, this is not true for multiple imputation (MI). Therefore, GMM calibration and MLE appear to use automatic classifications more efficiently than MI does.

In brief, when misclassifications cause nondifferential error, our simulations provide evidence that MLE and GMM calibration are both effective, efficient and provide accurate uncertainty quantification. These two methods complement each other since they have different assumptions and advantages. In theory, MLE depends on correctly specifying the likelihood and its robustness to incorrect specifications is difficult to analyze (Carroll et al., 2006). GMM calibration depends on an exclusion restriction instead of such distributional assumptions (Fong & Tyler, 2021). As discussed above, MLE's advantages over GMM calibration come from the relative ease with which it can be extended to more complex statistical models such as generalized linear models (GLMs) and generalized additive models (GAMs). Therefore, in cases similar to *Simulation 1a* we recommend using both GMM and an appropriately specified MLE model.

Simulation 1b: When Misclassifications Depend on the Outcome

Differential error can give rise to dramatic bias that is more difficult to correct using measurement error methods. As Figure 3 shows, the naïve estimator is opposite in sign to the true parameter in *Simulation 1b*. Of the four methods we test, only the MLE and the MI approach provide consistent estimates. This is expected because these are the only two methods using the outcome Y to adjust for errors in classifications. The bottom row of Figure 3 shows how the precision of the MI and MLE estimates increase with additional unlabeled data. As with *Simulation 1a*, MLE uses this data more efficiently than MI does. However, due to the low accuracy and bias of the AC, additional unlabeled data improves precision less than one might expect. Both methods provide acceptably accurate confidence intervals. Figure E2 in the Supplement shows that as in *Simulation 1a*, effective correction for misclassifications of X is required to consistently estimate B_Z, the coefficient of Z on Y. Looking at results from methods that do not correct differential error is useful for understanding their limitations. When few true values of X are known, GMM is nearly as bad as the naïve estimator, and PL is also visibly biased. Both improve when a greater proportion of the entire dataset is labeled because they combine their AC-based estimates with the feasible estimator.

In sum, our simulations suggest that the MLE method is the superior choice when misclassifications are not conditionally independent of the outcome given observed covariates. Although MI estimations are consistent, the method's practicality is limited by its inefficiency.

Simulation 2a: When Random Misclassifications Affect the Outcome

Ignoring misclassification in dependent variables also introduces bias as evidenced by the naïve estimator's inaccuracy illustrated in Figure 4. Both our MLE method and MI are able to correct this error and provide consistent estimates, but MLE is more efficient. It is puzzling that the MI estimator is inconsistent and does not improve with more human-labeled data. The PL approach is also inconsistent, especially when the validation dataset is small compared to the entire dataset, but it is closer to recovering the true parameter than the MI or naïve estimators. Based on Figure 4, it is clear that the precision of the MLE estimator improves with the addition of unlabeled data to a greater extent than the PL estimator. The PL estimator provides only modest improvements in precision compared to the feasible estimator. When the amount of human-labled data is low, inaccuracies in the 95% confidence intervals of both the MLE and PL become visible. As before, PL's inaccurate confidence intervals are due to its use of finite-sample estimates of the automatic classification probabilities.

In brief, our simulations suggest that MLE is the best of the methods we tested when misclassifications affect the dependent variable. It is the only consistent option and more efficient than the PL method, which is almost consistent.

Simulation 2b: When Misclassifications Affecting the Outcome Are Biased

In *Simulation 2b*, misclassifiations in the outcome are correlated with a covariate X. As shown in Figure 5, this type of misclassification can cause dramatic bias in the naïve estimator. Similar to *Simulation 1a*, MI is inconsistent, however PL is also inconsistent because it does not account for X in its measurement error model. As in *Simulation 1b*, our MLE method obtains consistent estimates, but only does much better than the feasible estimator when the dataset is large. Figure E in the Supplement shows the precision of estimates for the coefficient for X improves with additional data to a greater extent and so this imprecision is mainly in estimating the coefficient for Z, the variable correlated with misclassification. Therefore, our simulations suggest that MLE is the best method when misclassifications in the outcome are correlated with a covariate.

Transparency Is Not Enough. We Can Fix It!: Recommendations for Automated Content Analyses

“Validate, Validate, Validate” (Grimmer & Stewart, 2013) is one of the guiding mantras for automated content analysis. It reminds us that ACs can produce misleading results and of the importance of steps to ascertain their validity, for instance by making misclassificition rates transparent. Grimmer and Stewart (2013, p.5) write that “when categories are known [...], scholars must demonstrate that the supervised methods are able to reliably replicate human coding.” This suggests that quantifying an AC’s predictive performance by comparing human-labeled validation data to automatic classifications sufficiently establishes an AC’s validity and thereby the validity of downstream analyses.

Like Grimmer and Stewart (2013), we are deeply concerned that computational methods may produce invalid evidence. In this sense, their validation mantra animates this paper. But transparency about misclassification rates via metrics such as precision or recall leaves unanswered an important question: Is comparing automated classifications to external ground truth sufficient to claim validity? Or is there something else we can do and should do? We think there is: Using statistical methods to not only quantify but also correct for misclassification. Our study provides several recommendations in this regard.

Construct Validation Data before Building an AC

Analyzing human-coded data for validation is often done *post facto*, e.g., to calculate predictiveness metrics an AC is built. We propose to instead to collect and use manually annotated validation data *ante facto*. Practically speaking, the main reason to use an AC is feasibility, i.e., avoiding to label large data sets manually. For example, a large dataset may be necessary to study a small effect xand manually labeling such a dataset may be more expensive than building an AC. In this way, ACs can be seen as a cost-saving procedure that exchanges the expense of manual labeling in exchange for the threats to validity posed by misclassification. However, building an AC can also be very expensive because of the considerable costs of human annotation, software development, and computational resources needed to train ACs. Due to this often unpredictable effort, we caution researchers against building an AC unless doing so is necessary to obtain useful evidence. Instead, validation data should be used *ante facto*, with researchers beginning with preliminary analysis of human-coded data from which they can discern if an AC is necessary. In our simulations, the "feasible estimate" is less precise but consistent in all cases. So if fortune shines and this estimate sufficiently answers one's research question, the costs of building the AC are avoided. If feasible estimation fails to provide convincing evidence, for example by not rejecting a null hypothesis, the human-labeled data is not wasted. It can be reused to validate the AC and account for misclassification in downstream analysis.

Use Validation Data to Evaluate Differential Error

As we argued and demonstrated in our simulations, biases introduced by misclassification may not be trivial to adjust. Here, knowing whether an AC makes differential misclassification is particularly important for downstream analysises: It determines which correction method might work best. Fortunately, human coded data can be used to investigate differential misclassification. For example, "algorithmic audits" (e.g. Kleinberg et al., 2018; Rauchfleisch & Kaiser, 2020) evaluate the performance of AC across different subgroups in the data for example when using AC for corpora of different languages or data from different social media platforms. Differential misclassification can be ruled out if the performance is the same across all analytically relevant subgroups and other variables. We strongly recommend using such methods to test for differential misclassification and design the measurement error model within our MLE framework. Evidence that the model effectively corrects differential error can be provided by tests of conditional independence between the automatic classifications W and the outcome Y given a chosen model of $P(W|Y, X, Z)$, the conditional probability of the automatic classifications given the outcome and covariates.

Correct for Misclassification Errors (Twice) Instead of Being Naïve

Across our simulations, we showed that the naïve estimator is biased. Testing different error correction methods, we found that these generate different levels of consistency, efficiency, and accuracy in uncertainty quantification. That said, our proposed MLE method should be considered as a versatile method because it is the only method capable of producing consistent estimates in prototypical situations studied here. We recommend the MLE method as the first "go-to" method. The method requires specifying the error model, but this can be known if one follows our second recommendation. We developed the **misclassificationmodels** R package to facilitate adoption of our MLE method (see Appendix D in our Supplement).

We recommend comparing our MLE approach to another error correction method. Consistency between two correction methods shows that results are robust independent of the choice of correction method. If the AC is used to predict the dependent variable, PL might be a reasonable choice. For cases of AC-predicted covariates, GMM calibration is a good choice if error is nondifferential. Otherwise, MI can be considered. The range of viable choices in error correction motivates our next recommendation.

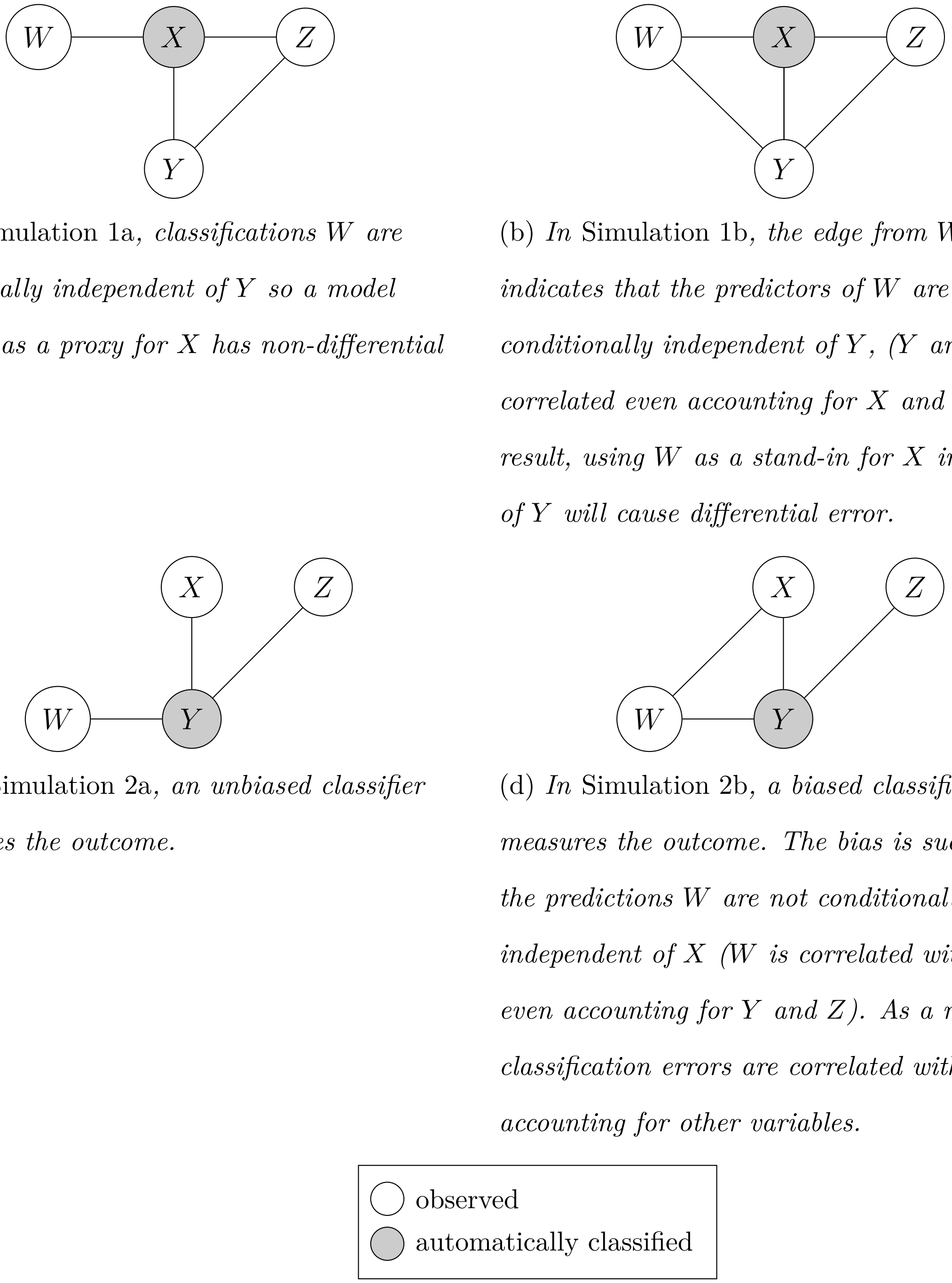
Provide a Full Account of Methodological Decisions and Robustness Checks

Finally, we add our voices to those recommending that researchers report methodological decisions so other can understand and replicate their design (Pipal et al., 2022; Reiss et al., 2022). These decisions include but are not limited to choices concerning test and training data (e.g., size, sampling, split in cross-validation procedures, balance), manual annotations (size of manually annotated data, number of coders, intercoder values, size of data coded for intercoder testing), and the classifier itself (choice of algorithm or ensemble, different accuracy metrics). They extend to reporting different error correction methods as proposed by our third recommendation. In our review, we found that reporting such decisions is not yet common, at least in the context of SML-based text classification. When correcting for misclassification, uncorrected results will often provide a lower-bound on effect sizes; corrected analyses will provide more accurate but less conservative results. Therefore, both corrected and uncorrected estimates should be presented as part of making potential multiverses of findings transparent. We realize that researchers might need to cut methodological information, especially for empirical studies, to conform to either word limits or reviewers. If word limitations are the problem, this information could be reported in appendices.

Conclusion and Limitations

We introduced the often-ignored problem of misclassification in automated content analysis, a topic often discussed in the context of manual content analysis (Scharkow & Bachl, 2017), but that we believe has not attracted enough attention within the computational social science community. In a systematic review of SML applications, we show that scholars rarely acknowledge this problem. We therefore discuss a range of statistical methods that use manually annotated validation data as a “gold standard” to account for misclassification and produce correct statistical results, including a new MLE method we design. Using Monte-Carlo simulations, we show that our method provides consistent estimates, especially in less trivial situations involving differential error. Based on these results, we provide four recommendations for the future of automated content analysis: Researchers should (1) construct manually annotated validation data before running ACs to see whether using human-labeled data is sufficient, (2) use validation data to test for differential error and choose error correction methods (3) correct for misclassifications via more than one error correction method, and (4) be transparent about the methodological decisions involved in SML-based classifications and error correction.

Our study has several limitations. First, the simulations and methods we introduce focus on misclassification by automated tools. They provisionally assume that human coders do not make errors. This assumption can be reasonable if intercoder reliability is very high but this may not always be the case. Thus, it may be important to account for measurement error by human classifiers and by automatic classifiers simultaneously. In theory, it is possible to extend our MLE approach in order to do so (Carroll et al., 2006). However, because the true values of content categories are never observed, accounting for automatic and human misclassification at once requires latent variable methods that bear considerable additional complexity and assumptions (Pepe & Janes, 2007). We leave the integration of such methods into our MLE framework for future work. Second, the simulations we present do not consider a number of factors that may influence the performance and robustness of the methods we test including classifier accuracy, heteroskedasticity, and violations of distributional assumptions. We are working to investigate such factors by extending our simulations. We simulated datasets with balanced covariates, but classifiers are often used to measure rare occurrences. Imbalanced covariates will require greater sample sizes of validation data to correct misclassification bias. In such cases, validation data may be collected more efficiently using approaches that provide balanced, but unrepresentative samples. Such non-representative sampling requires correction methods to account for probability that a datapoint will be sampled, but we have not evaluated if the correction methods can do so.



(a) *In* Simulation 1a*, classifications* W *are conditionally independent of* Y *so a model using* W *as a proxy for* X *has non-differential error.*

(b) *In* Simulation 1b*, the edge from* W *to* Y *indicates that the predictors of* W *are not conditionally independent of* Y*, (*Y *and* W *are correlated even accounting for* X *and* Z*). As a result, using* W *as a stand-in for* X *in a model of* Y *will cause differential error.*

(c) *In* Simulation 2a*, an unbiased classifier measures the outcome.*

(d) *In* Simulation 2b*, a biased classifier measures the outcome. The bias is such that the predictions* W *are not conditionally independent of* X *(*W *is correlated with* X *even accounting for* Y *and* Z*). As a result, classification errors are correlated with* Y *even accounting for other variables.*

Figure 1

Bayesnet networks representing the conditional independence structure of our simulations.

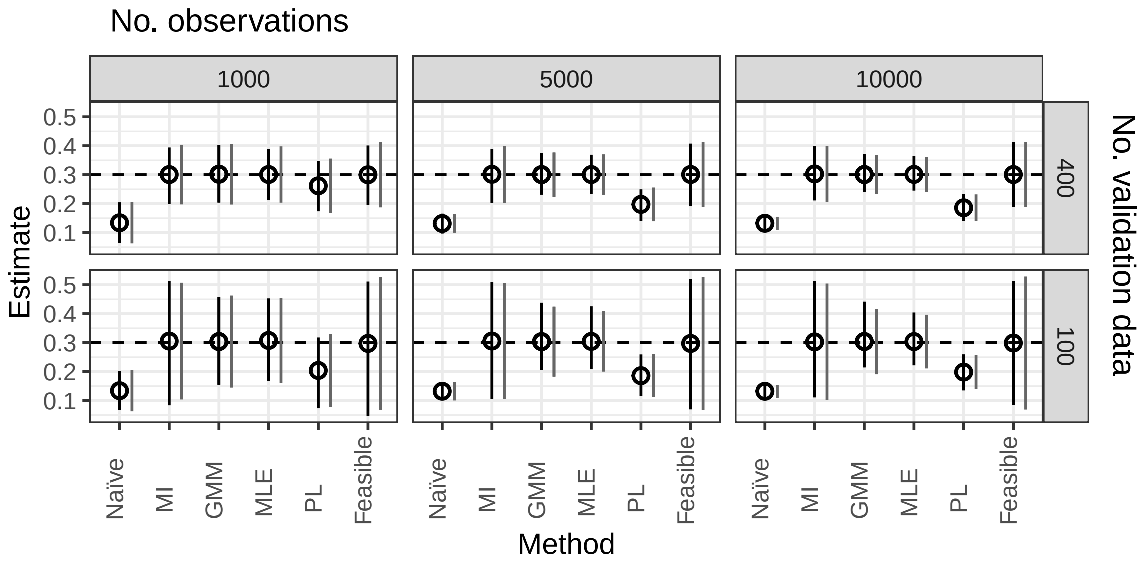


Figure 2

Estimates of B_X in multivariate regression with X measured using machine learning and model accuracy independent of X, Y, and Z. All methods, except the pseudo-likelihood method obtain precise and accurate estimates given sufficient validation data.

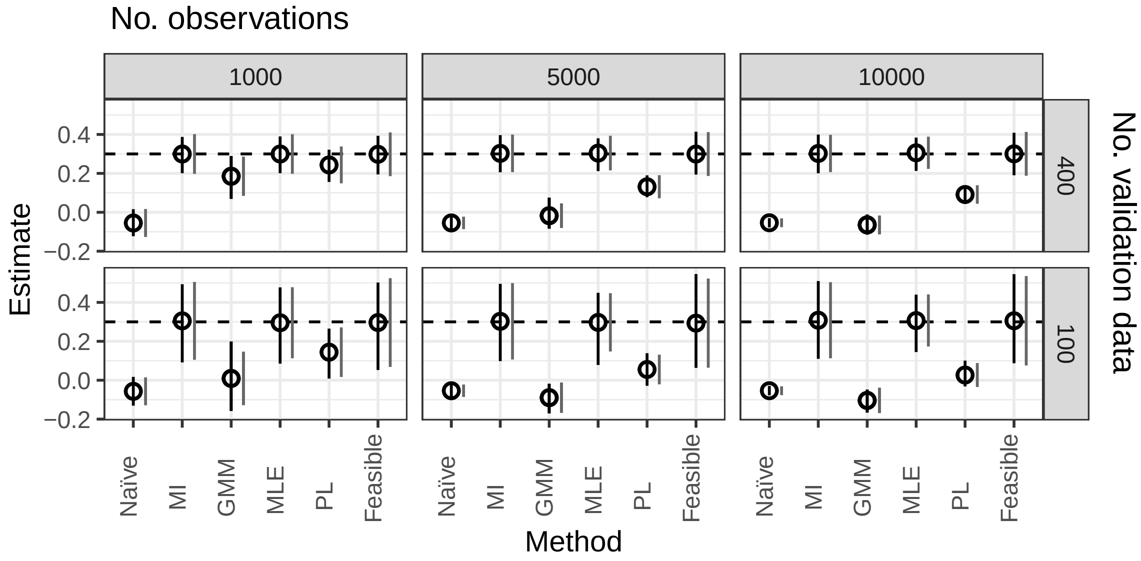


Figure 3

Estimates of B_X in multivariate regression with X measured using machine learning, where model accuracy correlated with X and Y. Only multiple imputation and our MLE model with a full specification of the error model obtain consistent estimates of B_X.

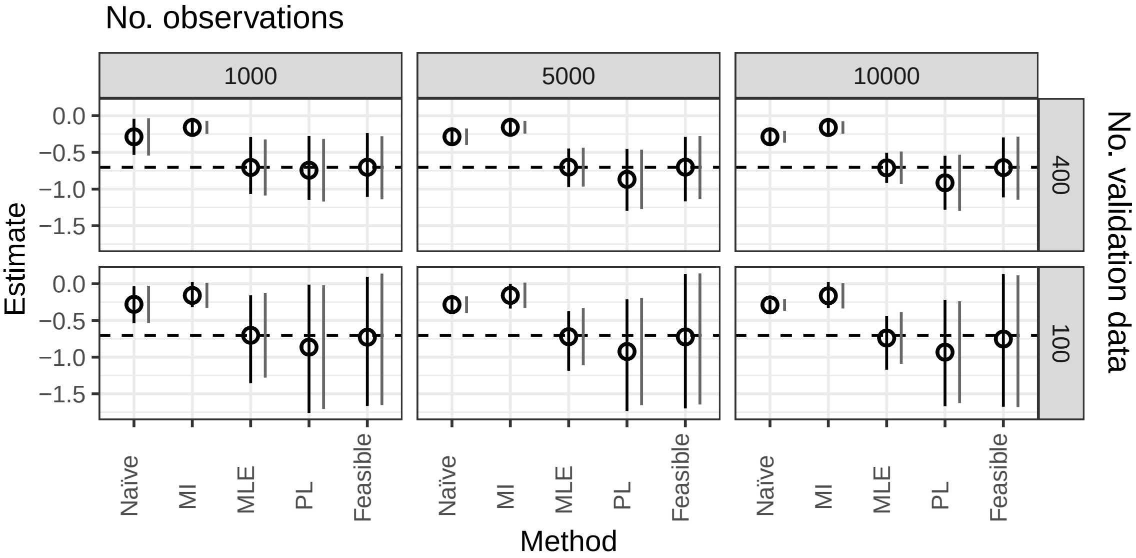


Figure 4

Estimates of B_Z in Simulation 1b*, multivariate regression with Y measured using an imperfect automatic classifier. Only our MLE model obtains consistent estimates .*

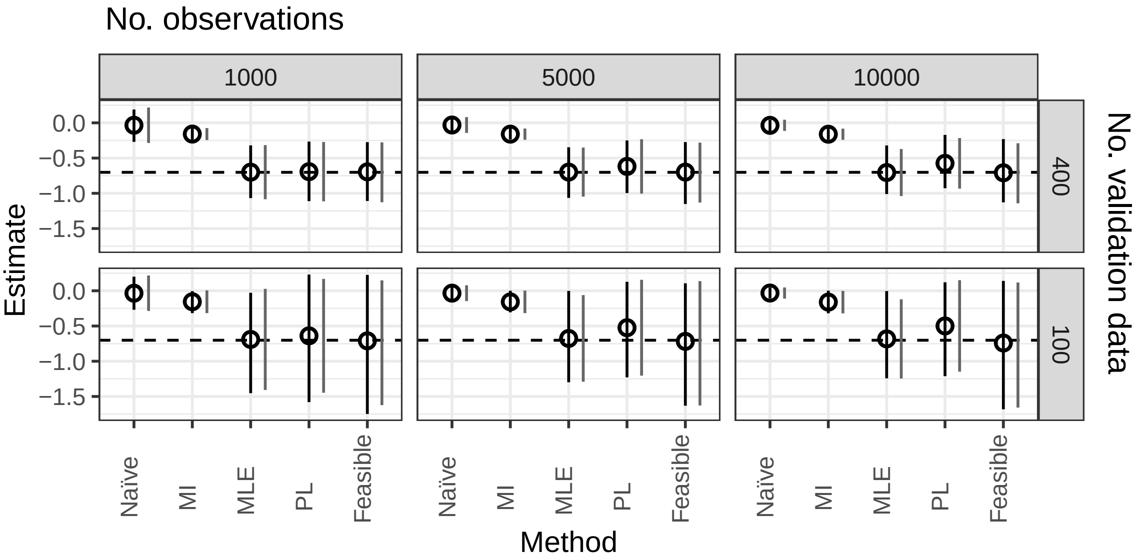


Figure 5

Estimates of B_Z in Simulation 2b*, multivariate regression with Y measured using an automatic classifier that makes errors correlated a covariate X. Only our MLE model with a full specification of the error model obtains consistent estimates.*

References

Bachl, M., & Scharkow, M. (2017). Correcting Measurement Error in Content Analysis. *Communication Methods and Measures*, *11*(2), 87–104.

Baden, C., Pipal, C., Schoonvelde, M., & van der Velden, M. A. C. G. (2022). Three Gaps in Computational Text Analysis Methods for Social Sciences: A Research Agenda. *Communication Methods and Measures*, *16*(1), 1–18.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623.

Blackwell, M., Honaker, J., & King, G. (2012). Multiple Overimputation: A Unified Approach to Measurement Error and Missing Data, 50.

Boukes, M., van de Velde, B., Araujo, T., & Vliegenthart, R. (2020). What's the Tone? Easy Doesn't Do It: Analyzing Performance and Agreement Between Off-the-Shelf Sentiment Analysis Tools. *Communication Methods and Measures*, *14*(2), 83–104 _eprint: https://doi.org/10.1080/19312458.2019.1671966.

Breiman, L. (2001). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

Buonaccorsi, J. P. (2010, July 19). *Measurement Error: Models, Methods, and Applications*. Chapman and Hall/CRC.

Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. M. (2006). *Measurement Error in Nonlinear Models* (2nd ed.). Chapman & Hall/CRC.

Fong, C., & Tyler, M. (2021). Machine Learning Predictions as Regression Covariates. *Political Analysis*, *29*(4), 467–484.

Fuller, W. A. (1987). *Measurement error models*. Wiley.

Geiß, S. (2021). Statistical Power in Content Analysis Designs: How Effect Size, Sample Size and Coding Accuracy Jointly Affect Hypothesis Testing – A Monte Carlo Simulation Approach. *Computational Communication Research*, *3*(1), 61–89.

González-Bailón, S., & Paltoglou, G. (2015). Signals of Public Opinion in Online Communication: A Comparison of Methods and Data Sources. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 95–107.

Grimmer, J., & Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, *21*(3), 267–297.

Hase, V., Mahl, D., & Schäfer, M. S. (2022). Der „Computational Turn“: Ein „interdisziplinärer Turn“? Ein systematischer Überblick zur Nutzung der automatisierten Inhaltsanalyse in der Journalismusforschung. *Medien & Kommunikationswissenschaft*, *70*(1-2), 60–78.

Hede, A., Agarwal, O., Lu, L., Mutz, D. C., & Nenkova, A. (2021, February 6). *From Toxicity in Online Comments to Incivility in American News: Proceed with Caution.*

Hopkins, D. J., & King, G. (2010). A Method of Automated Nonparametric Content Analysis for Social Science. *American Journal of Political Science*, *54*(1), 229–247.

Hosseini, H., Kannan, S., Zhang, B., & Poovendran, R. (2017, February 26). Deceiving Google's Perspective API Built for Detecting Toxic Comments.

Jünger, J., Geise, S., & Hännelt, M. (2022). Unboxing Computational Social Media Research From a Datahermeneutical Perspective: How Do Scholars Address the Tension Between Automation and Interpretation? *International Journal of Communication*, *16*, 1482–1505.

Kleinberg, J., Ludwig, J., Mullainathan, S., & Rambachan, A. (2018). Algorithmic Fairness. *AEA Papers and Proceedings*, *108*, 22–27.

Krippendorff, K. (2004). Reliability in Content Analysis. *Human Communication Research*, *30*(3), 411–433

_eprint:

https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1468-2958.2004.tb00738.x.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. SAGE.

Loken, E., & Gelman, A. (2017). Measurement error and the replication crisis. *Science*, *355*(6325), 584–585.

Lovejoy, J., Watson, B. R., Lacy, S., & Riffe, D. (2014). Assessing the Reporting of Reliability in Published Content Analyses: 1985–2010. *Communication Methods and Measures*, *8*(3), 207–221.

Millimet, D. L., & Parmeter, C. F. (2022). Accounting for Skewed or One-Sided Measurement Error in the Dependent Variable. *Political Analysis*, *30*(1), 66–88.

Mooney, C. Z. (1997). *Monte Carlo simulation*. Sage Publications, Inc.

Opperhuizen, A. E., Schouten, K., & Klijn, E. H. (2019). Framing a Conflict! How Media Report on Earthquake Risks Caused by Gas Drilling: A Longitudinal Analysis Using Machine Learning Techniques of Media Reporting on Gas Drilling from 1990 to 2015. *Journalism Studies*, *20*(5), 714–734.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, *29*(3), 241–288.

Pepe, M. S., & Janes, H. (2007). Insights into latent class analysis of diagnostic test performance. *Biostatistics*, *8*(2), 474–484.

Pipal, C., Song, H., & Boomgaarden, H. G. (2022). If You Have Choices, Why Not Choose (and Share) All of Them? A Multiverse Approach to Understanding News Engagement on Social Media. *Digital Journalism*, 1–21.

Rauchfleisch, A., & Kaiser, J. (2020). The False positive problem of automatic bot detection in social science research. *PLOS ONE*, *15*(10), e0241045.

Reiss, M., Kobilke, L., & Stoll, A. (2022, June 10). *Reporting Supervised Text Analysis for Communication Science*. Munich.

Rice, D., Siddiki, S., Frey, S., Kwon, J. H., & Sawyer, A. (2021). Machine coding of policy texts with the Institutional Grammar. *Public Administration*, *99*(2), 248–262 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/padm.12711.

Scharkow, M. (2017). Content analysis, automatic. *The international encyclopedia of communication research methods*, 1–14.

Scharkow, M., & Bachl, M. (2017). How Measurement Error in Content Analysis and Self-Reported Media Use Leads to Minimal Media Effect Findings in Linkage Analyses: A Simulation Study. *Political Communication*, *34*(3), 323–343 _eprint: https://doi.org/10.1080/10584609.2016.1235640.

Song, H., Tolochko, P., Eberl, J.-M., Eisele, O., Greussing, E., Heidenreich, T., Lind, F., Galyga, S., & Boomgaarden, H. G. (2020). In Validations We Trust? The Impact of Imperfect Human Annotations as a Gold Standard on the Quality of Validation of Automated Content Analysis. *Political Communication*, *37*(4), 550–572.

van Atteveldt, W., van der Velden, M. A. C. G., & Boukes, M. (2021). The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms. *Communication Methods and Measures*, *15*(2), 121–140 _eprint: https://doi.org/10.1080/19312458.2020.1869198.

van Smeden, M., Lash, T. L., & Groenwold, R. H. H. (2020). Reflection on modern methods: Five myths about measurement error in epidemiological research. *International Journal of Epidemiology*, *49*(1), 338–347.

Vermeer, S., Trilling, D., Kruikemeier, S., & de Vreese, C. (2020). Online News User Journeys: The Role of Social Media, News Websites, and Topics. *Digital Journalism*, *8*(9), 1114–1141.

Williams, C., & Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1342–1351.

Yi, G. Y., Delaigle, A., & Gustafson, P. (Eds.). (2021, October 17). *Handbook of Measurement Error Models*. Chapman and Hall/CRC.

Zhang, H. (2021, May 29). *How Using Machine Learning Classification as a Variable in Regression Leads to Attenuation Bias and What to Do About It* (preprint). SocArXiv.

Appendix A

Systematic Literature Review

To understand scholarly awareness of measurement errors, we conducted a systematic literature review of common practices in SML-based text classification.

Identification of Relevant Studies

To identify relevant studies, we relied on four recent reviews on the use of AC with a focus on communication science (Baden et al., 2022; Hase et al., 2022; Jünger et al., 2022; Song et al., 2020). We contacted authors of respective studies who, thankfully, either already published their data in an open-science approach or shared their data with us when asked. Based on their reviews, we collected $N = 110$ studies that, according to their analyses, included some type of SML (for an overview, see Figure A1).

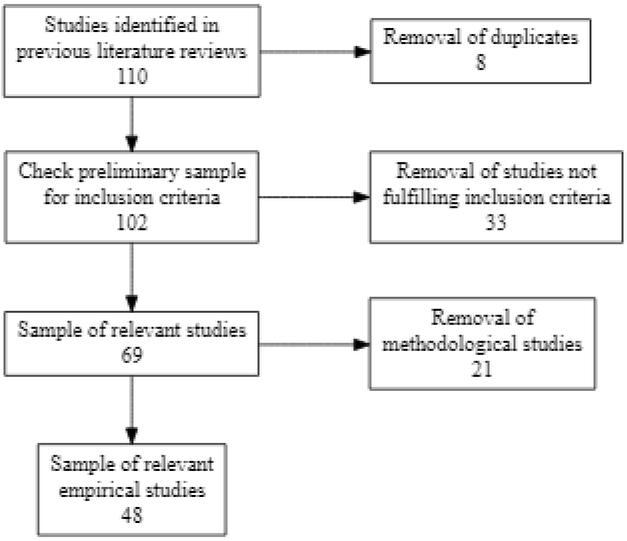

Studies identified in previous literature reviews
110
Removal of duplicates
8
Check preliminary sample for inclusion criteria
102
Removal of studies not fulfilling inclusion criteria
33
Sample of relevant studies
69
Removal of methodological studies
21
Sample of relevant empirical studies
48

Figure A1

Identifying relevant studies for the literature review

We first removed 8 duplicate studies identified by several reviews. Two coders then coded the remaining $N = 102$ studies of our preliminary sample for relevance. After an intercoder test ($N = 10$, $\alpha = .89$), coders sorted studies into one of four categories: Similar to previous reviews (Hase et al., 2022), we only included studies either focusing on methodologically advancing SML-based ACs (Code = 1) or applying the method in empirical studies (Code = 2). In contrast, we removed studies that did not include any SML approach (Code = 3) or only used SML-based text classification for data cleaning, not data analysis (Code = 4)—for instance to sort out topically irrelevant articles.

Subsequently, $N = 69$ studies remained in our sample of relevant articles. Out of these, only empirical studies ($N = 48$) were coded in further detail. We explicitly excluded methodological studies for understanding common practices within SML-based text classification since these will like include far more robustness and validity tests than commonly employed in empirical settings.

Manual Coding of Relevant Empirical Studies

For the remaining $N = 48$ empirical studies, we created a range of variables (for an overview, see Table A1). Based on data from the Social Sciences Citation Index (SSCI), we identified whether studies were published in journals classified as belonging to *Communication* and their *Impact* according to their H index. In addition, two coders manually coded...

- the type of variables created via SML-based ACS using the variables *Dichotomous* (0 = No, 1 = Yes), *Categorical* (0 = No, 1 = Yes), *Ordinal* (0 = No, 1 = Yes), *Metric* (0 = No, 1 = Yes),
- whether variables were used in descriptive or multivariate analyses using the variables *Descriptive* (0 = No, 1 = Yes), *Independent* (0 = No, 1 = Yes), *Dependent* (0 = No, 1 = Yes),
- how classifiers were trained and validated via manually annotated data using the variables *Size Training Data* (Open String), *Size Test Data* (Open String), *Size Data Intercoder Test* (Open String), *Intercoder Reliability* (Open String), *Accuracy of Classifier* (Open String),
- and whether articles mentioned and/or corrected for misclassifications using the

variables *Error Mentioned* (0 = No, 1 = Yes) and *Error Corrected*) (0 = No, 1 = Yes).

Table A1

Variables Coded for Relevant Empirical Studies

Category	Variable	Krippendorf's α	% or *M* (*SD*)
Type of Journal	*Communication*	n.a.	55.1%
	Impact	n.a.	*M = 3.69*
Type of Variable	*Dichotomous*	0.86	50%
	Categorical	1	22.9%
	Ordinal	0.85	10.4%
	Metric	1	35.4%
Use of Variable	*Descriptive*	0.89	89.6%
	Independent	1	43.8%
	Dependent	1	39.6%
Information on Classifier	*Size Training Data*	0.95	66.7%
	Size Test Data	0.79	52.1%
	Size Data Intercoder Test	1	43.8%
	Intercoder Reliability	0.8	56.2%
	Accuracy of Classifier	0.77	85.4%
Measurement Error	*Error Mentioned*	1	18.8%
	Error Corrected	1	2.1%

Results

Overall, more than half of all studies were published in communication journals (*Communication*: 55.1%). Across domains, SML-based ACs were most often used to create dichotomous measurements (*Dichotomous*: 50%), followed by variables on a metric (*Metric*:

35.4%), categorical (*Categorical*: 22.9%), or ordinal scale (*Ordinal*: 10.4%). Almost all studies used SML-based classifications to report descriptive statistics on created variables (*Descriptive*: 89.6%). However, many also used these in downstream analyses, either as dependent variables (*Dependent*: 39.6%) or independent variables (*Independent*: 43.8%) in multivariate models. When regressing the use of multivariate models for each variable on the status of journals in which respective studies were published *Impact*) via a mixed model where variables are nested in studies and journals, we find that both correlate: The use of multivariate modeling is more widespread in high-impact journals ($B = 13.525$, $p < .001$)

Overall, we found a persistent lack of transparency in reporting important information: Only slightly more than half of all studies included information on, for instance, the size of training or test sets (*Size Training Data*: 66.7%, *Size Test Data*: 52.1%). Even fewer included information on the size of manually annotated data for intercoder testing (*Size Data Intercoder Test*: 43.8%) or respective reliability values (*Intercoder Reliability*: 56.2%). Lastly, not all studies reported how well their classifier performed by using metrics such as precision, recall, or F1-scores (*Accuracy of Classifier*: 85.4%).

Lastly, we also found that few studies mentioned the issue of misclassification or measurement errors (*Error Mentioned*: 18.8%, with only a single study correcting for such (*Error Corrected*: 2.2%).

Appendix B

Other methods not tested

Simulation extrapolation (SIMEX) uses a simulation of the process generating measurement error to model how measurement error affects an analysis and ultimately to approximate an analysis with no measurement error (Carroll et al., 2006). SIMEX is a very powerful and general method that can be used without validation data, but may be more complicated than necessary to correct measurement error from ACs when validation data are available. Likelihood methods are easy to apply to classification errors so SIMEX seems unnecessary (Carroll et al., 2006).

Score function methods derive estimating equations for models without measurement error and then solve them either exactly or using numerical integration (Carroll et al., 2006; Yi et al., 2021). The main advantage of score function methods may have over likelihood-based methods is that they do not require distributional assumptions about the mismeasured covariates. This advantage has limited use in the context of ACs because binary classifications must follow Bernoulli distributions.

We also do not consider Bayesian methods (aside from the Amelia implementation of multiple imputation) because we expect these to have similar limitations to the maximum likelihood methods we consider. Bayesian methods may have other advantages resulting from posterior inference, and may generalize to a wide range of applications, but specifying prior distributions introduces additional methodological complexity and posterior inference is computationally intensive making Bayesian methods less convenient for monte-carlo simulation.

Appendix C

Deriving the maximum likelihood approach

When an AC measures a covariate

To show why $L(\theta|Y, W)$ can be factored, we follow Carroll et al. (2006) and begin by observing the following fact from basic probability theory.

$$P(Y, W) = \sum_x P(Y, W, X = x) \tag{C1}$$
$$= \sum_x P(Y|W, X = x)P(W, X = x) \tag{C2}$$
$$= \sum_x P(Y, X = x)P(W|Y, X = x) \tag{C3}$$
$$= \sum_x P(Y|X = x)P(W|Y, X = x)P(X = x) \tag{C4}$$

Equation C1 integrates X out of the joint probability of Y and W by summing over its possible values x. If X is binary, this means adding the probability given $x = 1$ to the probability given $x = 0$. When X is observed, say $x = 0$, then $P(X = 0) = 1$ and $P(X = 1) = 0$. As a result, only the true value of X contributes to the likelihood. However, when X is unobserved, all of its possible values contribute. In this way, integrating out X allows us to include data where X is not observed to the likelihood.

Equation C2 uses the chain rule of probability to factor the joint probability $P(Y, W)$ of Y and W from $P(Y|W, X)$, the conditional probability of Y given W and X and $P(W, X = x)$, the joint probability of W and X. This lets us see how maximizing $\mathcal{L}(\Theta|Y, W)$, the joint likelihood of Θ given Y and W accounts for the uncertainty of the automatic classifications. For each possible value x of X, it weights the model of the outcome Y by the probability that x is the true value and that the AC outputs W.

Equation C3 shows a different way to factor the joint probability $P(Y, W)$ so that W is not in model of Y. Since X and W are correlated, if W in the model for Y estimation of B_1 will be biased. By including Y in the model for W, Equation C3 can account for differential measurement error.

Equation C4 factors $P(Y, X = x)$ the joint probability of Y and X into $P(Y|X = x)$, the conditional probability of Y given X, $P(W|X = x, Y)$, the conditional probability of W given X and Y, and $P(X = x)$ the probability of X. This shows that fitting a model Y given X, in this framework, such as the regression model $Y = B_0 + B_1X + B_2Z$ requires including X. Without validation data, $P(X = x)$ is difficult to calculate without strong assumptions (Carroll et al., 2006), but $P(X = x)$ can easily be estimated using a sample of validation data.

Equations C1–C4 demonstrate the generality of this method because the conditional probabilities may be calculated using a wide range of probability models. For simplicity, we proceed with a focus on linear regression for the probability of Y and logistic regression for the probability of W and the probability of X. However, more flexible probability models such as generalized additive models (GAMs) or Gaussian process classification may be useful for modeling nonlinear conditional probability functions (Williams & Barber, 1998).

When an AC measures the outcome

Again, we will maximize $\mathcal{L}(\Theta|Y, W)$, the joint likelihood of the parameters Θ given the outcome Y and the automatic classifications W measure the dependent variable Y (Carroll et al., 2006). Therefore, we use the law of total probability to integrate out Y and the chain rule of probability to factor the joint probability into $P(Y)$, the probability of Y, and $P(W|Y)$ as the conditional probability of W given Y.

$$P(Y, W) = \sum_y P(Y = y, W) \tag{C5}$$

$$= \sum_y P(Y)P(W|Y) \tag{C6}$$

As above, the conditional probability of W given Y must be calculated using a model. The range of possible models is vast and analysts must choose a model that accurately describes the conditional dependence of W on Y.

We implement these methods in `R` using the `optim` library for maximum likelihood estimation. Our implementation supports models specified using `R`'s formula syntax can fit linear and logistic regression models when an AC measures a covariate and logistic regression models when an AC measures the outcome. Our implementation provides two methods for approximating confidence intervals: The Fischer information quadratic approximation, and the profile likelihood method provided in the `R` package `bbmle`. The Fischer approximation usually works well in simple models fit to large samples and is fast enough for practical use for the large number of simulations we present. However, the profile likelihood method provides more accurate confidence intervals (Carroll et al., 2006).

Appendix D

misclassificationmodels: The R package

The package provides a function to conduct regression analysis but also corrects for misclassification in proxy using the information in validation data. The function is very simular to **glm()** but with two changes:

- The formula interface has been extended with the double-pipe operator to denote proxy variable. For example, **x || w** indicates w is the proxy of the ground truth x.
- The validation data must be provided

The following code listing shows a typical correction scenario:

```r
library(misclassificationmodels)
## research_data contains the following columns: y, w, z
## val_data contains the following columns: y, w, x, z
# w is a proxy of x
res <- glm_fixit(formula = y ~ x || w + z,
                 data = research_data,
                 data2 = val_data)
summary(res)
```

Listing 1: A demo of misclassificationmodels

Appendix E

Additional plots from Simulations 1 and 2

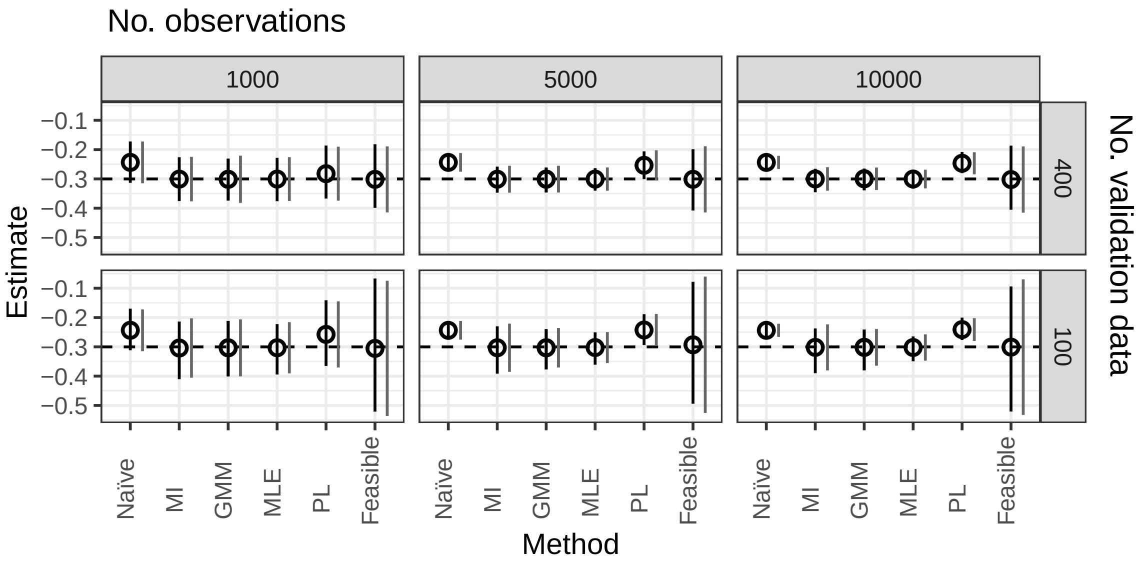


Figure E1

Estimates of B_Z in simulation 1a*, multivariate regression with X measured using machine learning and model accuracy independent of X, Y, and Z. All methods obtain precise and accurate estimates given sufficient validation data.*

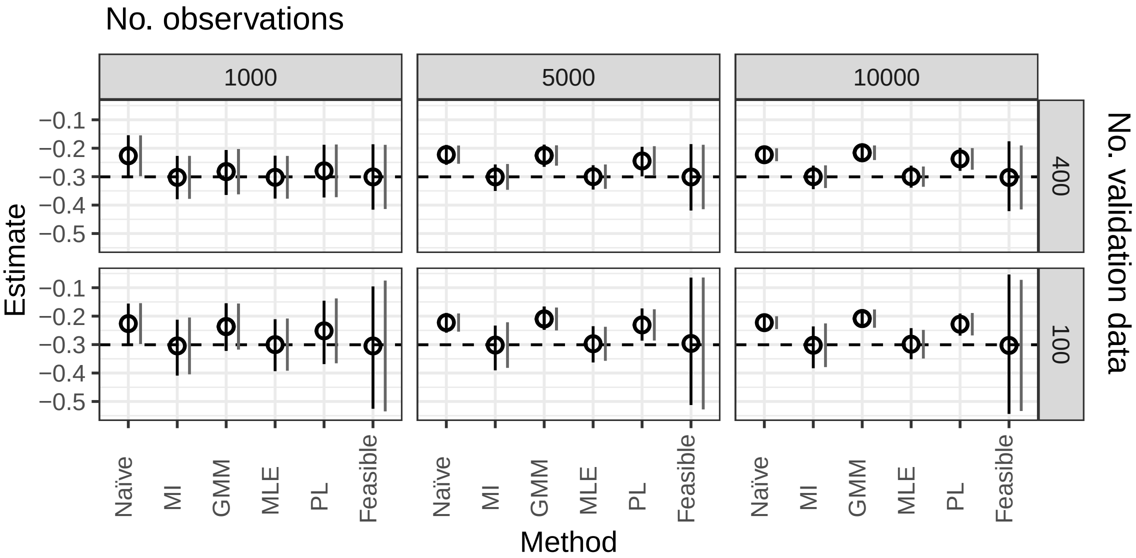


Figure E2

Estimates of B_Z in multivariate regression with X measured using machine learning and model accuracy correlated with X and Y and error is differential. Only multiple imputation and our MLE model with a full specification of the error model obtain consistent estimates of B_X.

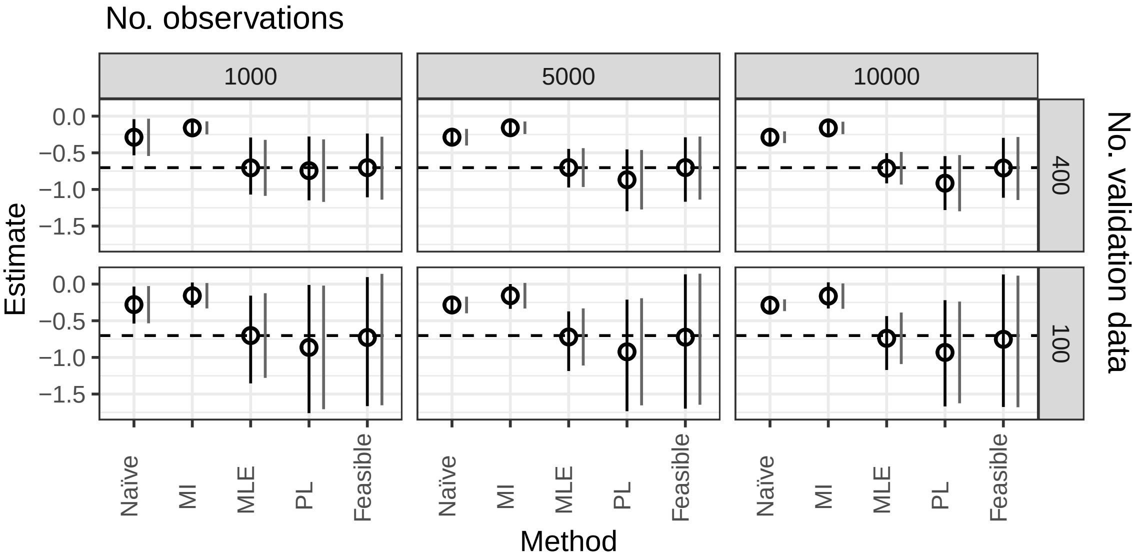


Figure E3

Estimates of B_Z in simulation 2a, *multivariate regression with Y measured using an AC that makes errors. Only our MLE model with a full specification of the error model obtains consistent estimates.*

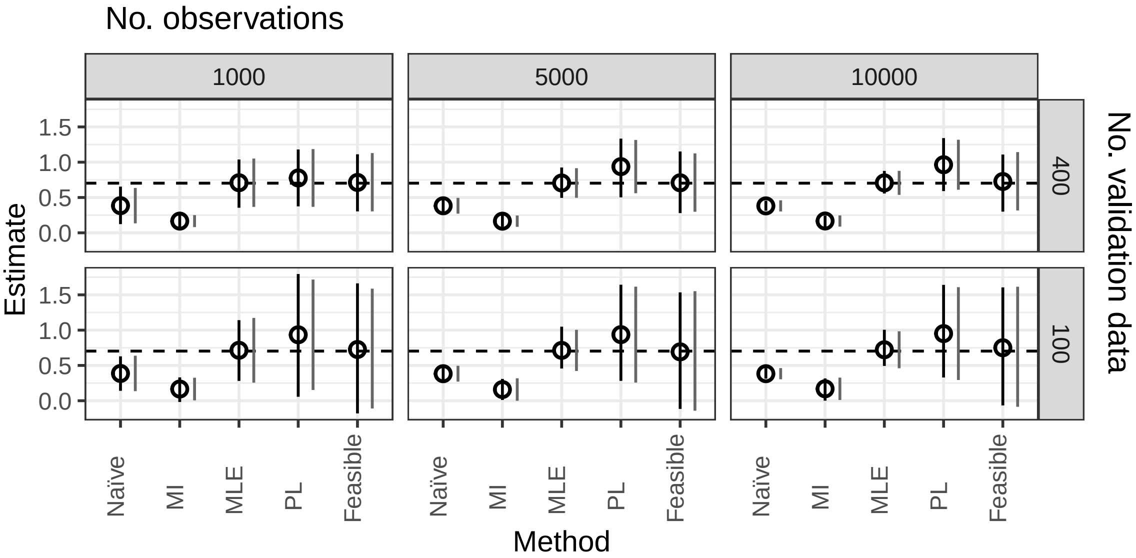


Figure E4

Estimates of B_X in simulation 2b *multivariate regression with Y measured using machine learning, model accuracy correlated with Z and Y and differential error. Only our MLE model with a full specification of the error model obtains consistent estimates.*