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

Nathan TeBlunthuis[1], Valerie Hase[2], and Chung-hong Chan[3]

[1]School of Information, University of Michigan,Department of Communication Studies, Northwestern University

[2]Department of Media and Communication, LMU Munich

[3]GESIS - Leibniz-Institut für Sozialwissenschaften

Abstract

We show how automated classifiers (ACs), even biased ACs without high accuracy, can be statistically useful in communication research. These classifiers, often built via supervised machine learning (SML), can categorize large, statistically powerful samples of data ranging from text to images and video, and have become widely popular measurement devices in communication science and related fields. Despite this popularity, even highly accurate classifiers make errors that cause misclassification bias and misleading results in downstream analyses—unless such analyses account for these errors. As we show in a systematic literature review of SML applications, communication scholars largely ignore misclassification bias. In principle, existing statistical methods can use "gold standard" validation data, such as that created by human annotators, to correct misclassification bias and produce consistent estimates. 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 our results, we provide recommendations for correcting misclassification bias. In sum, automated classifiers, even those below common accuracy standards or making systematic misclassifications, can be useful for measurement with careful study design and appropriate error correction methods.

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

Misclassification in Automated Content Analysis 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 ACs, researchers can categorize large samples of text, images, video or other types of data into predefined categories (Scharkow, 2013). Studies for instance use SML-based classifiers to study frames (Burscher et al., 2014), tonality (van Atteveldt et al., 2021), or civility (Hede et al., 2021) in news media texts or social media posts.

However, there is increasing concern about the validity of automated content analysis for studying theories and concepts from communication science (Baden et al., 2022; Hase et al., 2022). We add to this debate by analyzing *misclassification bias*—how misclassifications by ACs distort statistical findings—unless correctly modeled (Fong & Tyler, 2021). 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 bias (Rauchfleisch & Kaiser, 2020): How accurate must an AC be to measure a variable? Can an AC built for one context be used in another (Burscher et al., 2015; Hede et al., 2021)? Is comparing automated classifications to some external ground truth sufficient to claim validity? How do biases in AC-based measurements affect downstream statistical analyses (Millimet & Parmeter, 2022)?

Our study begins with a demonstration of misclassification bias in a real-world example based on the Perspective toxicity classifier. Next, we provide a systematic literature review of $N = 48$ studies employing SML-based text classification. Although communication scholars have long scrutinized related questions about manual content analysis for which they have recently proposed statistical corrections (Bachl & Scharkow, 2017; Geiß, 2021), misclassification bias in automated content analysis is largely ignored.

Our review demonstrates a troubling lack of attention to the threats ACs introduce and virtually no mitigation of such threats. As a result, in the current state of affairs, researchers are likely to 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).

Our primary contribution, an effort rescue ACs from this dismal state, is to *introduce and test methods for correcting misclassification bias* (Buonaccorsi, 2010; Carroll et al., 2006; Yi et al., 2021). We consider three recently proposed methods: Fong and Tyler (2021)'s generalized method of moments calibration method, Zhang (2021)'s pseudo-likelihood models, and Blackwell et al. (2017)'s application of imputation methods. To overcome these methods' limitations, we draw a general likelihood modeling framework from the statistical literature on measurement error (Carroll et al., 2006) and tailor it to the problem of misclassification bias. Our novel implementation is the experimental R package `misclassificationmodels`.[1]

We test these four error correction methods and compare them against ignoring misclassification (the naïve approach) and refraining from automated content analysis by only using manual coding (the feasible approach). We use Monte Carlo simulations to model four prototypical situations identified by our review: Using ACs to measure either (1) an independent or (2) a dependent variable where the classifier makes misclassifications that are either (a) easy to correct (when an AC is unbiased and misclassifications are uncorrelated with covariates i.e., *nonsystematic misclassification*) or (b) more difficult (when an AC is biased and misclassifications are correlated with covariates i.e., *systematic misclassification*).

According to our simulations, even biased classifiers without high predictive performance can be useful in conjunction with appropriate validation data and error

[1] The code for the experimental package can be found here: https://osf.io/pyqf8/?view_only=c80e7b76d94645bd9543f04c2a95a87e.

correction methods. As a result, we are optimistic about the potential of ACs and automated content analysis for communication science and related fields—if researchers correct for misclassification. Current practices of "validating" ACs by making misclassification rates transparent via metrics such as the F1 score, however, provide little safegaurd against misclassification bias.

In sum, we make a methodological contribution by introducing the often-ignored problem of misclassification bias in automated content analysis, testing error correction methods to address this problem via Monte Carlo simulations, and introducing a new method for error correction. Profoundly, we conclude that automated content analysis will progress not only—or even primarily—by building more accurate classifiers but by rigorous human annotation and statistical error modeling.

Why Misclassification is a Problem: an Example Based on the Perspective API

There is no perfect AC. All ACs make errors. This inevitable misclassification causes bias in statistical inference (Carroll et al., 2006; Scharkow & Bachl, 2017), leading researchers to make both type-1 (false discovery) and type-2 errors (failure to reject the null) in hypotheses tests. To illustrate the problematic consequences of this misclassification bias, we focus on real-world data and a specific research area in communication research: detecting and understanding harmful social media content. Communication researchers often employ automated tools such as the Perspective toxicity classifier (cjadams et al., 2019) to detect toxicity in online content (e.g., Hopp & Vargo, 2019; Kim et al., 2021; Votta et al., 2023). As shown next, however, relying on toxicity scores created by ACs such as the Perspective API as (in-)dependent variables produces different results than using measurements created via manual annotation.

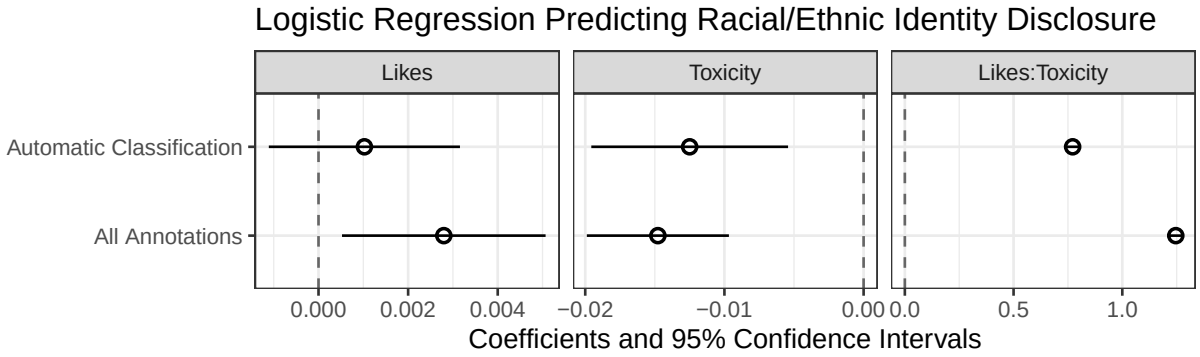
To illustrate this, we use the Civil Comments dataset released in 2019 by Jigsaw, the Alphabet corporation subsidiary behind the Perspective API. Methodological details on the data and our example are available in Appendix A. The dataset has 448,000 English-language comments made on independent news sites. It also includes manual

annotations of each comment concerning its toxicity (*toxicity*), whether it discloses aspects of personal identity like race or ethnicity *(identity disclosure)*, and the number of likes it received *(number of likes)*.

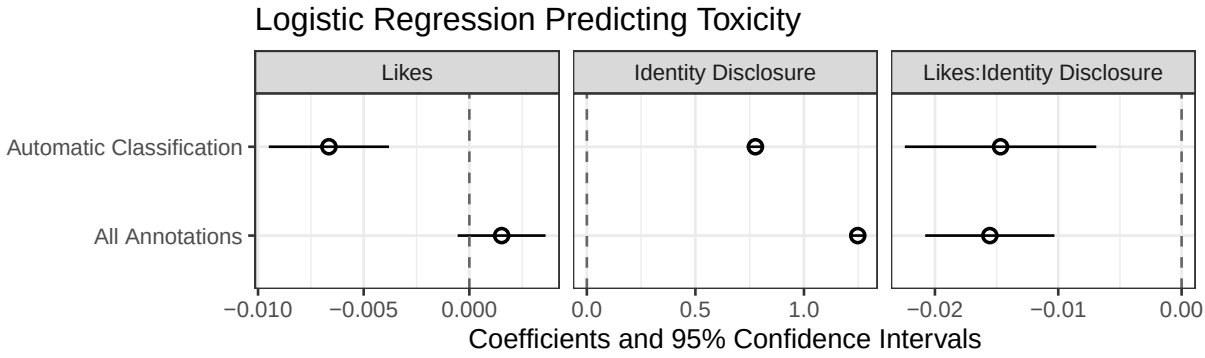
In addition to manual annotations of each comment, we obtained AC-based toxicity classifications from the Perspective API in November 2022. Perspective's toxicity classifier performs very well, with an accuracy of 92% and an F1 score of 0.79. Nevertheless, if we treat human annotations as the ground-truth, the classifier makes systematic misclassifications for it is modestly biased and disproportionately misclassifies comments disclosing racial or ethnic identity as toxic (Pearson's $\rho = 0.12$).

First, let us consider *misclassification in an independent variable*. As an example, we use a logistic regression model to predict whether a comment contains *identity disclosure* using *number of likes*, *toxicity*, and their interaction as independent variables. Although this is a toy example, it resembles a realistic investigation of how disclosing aspects of one's identity online relates to normative reception of one's behavior. As shown in Figure 1a, relying on AC-based toxicity classifications may lead researchers to reject a hypothesized direct relationship between likes and identity disclosure. Instead, the model suggests that their correlation is entirely mediated by toxicity. In contrast, using human annotations would lead researchers to conclude a subtle positive direct relationship between likes and identity disclosure. This demonstrates that even a very accurate AC can introduce type-2 error, i.e. researchers failing to rejecting a null hypothesis due to misclassification.

Second, let us consider *misclassification in a dependent variable*. We now predict the *toxicity* of a comment with *number of likes*, *identity disclosure* in a comment, and their interaction as independent variables. As shown in Figure 1b, using Perspective's classification of toxicity results in a small negative direct effect of likes. However, there is no detectable relationship when using manual annotations. As such, misclassification can also lead to type-1 error, i.e., false discovery of a nonzero relationship.



(a) Example 1*: Misclassification in an independent variable.*



(b) Example 2*: Misclassification in a dependent variable.*

Figure 1

Bias through Misclassification: a Real-World Example Using the Perspective API and the Civil Comments Dataset.

Why Transparency about Misclassification Is Not Enough

Although the Perspective API is no doubt accurate enough to be useful to content moderators, the example above demonstrates that this does not imply usefulness for social science (Grimmer et al., 2021). Machine learning takes the opposite position on the bias-variance trade-off than conventional statistics does and achieves high predictiveness at the cost of unbiased inference (Breiman, 2001). As a growing body of scholarship critical of the hasty adoption of machine learning in criminal justice, healthcare, or content moderation demonstrates, ACs boasting high performance often have biases related to social categories (Barocas et al., 2019). Such biases in machine learning often result from non-representative training data and spurious correlations that neither reflect causal mechanisms nor generalize to different populations (Bender et al., 2021). Much of this critique targets unjust consequences of these biases to individuals. Our example shows that these biases can also contaminate scientific studies using ACs as measurement devices. Even very accurate ACs can cause both type-I and type-II errors, which become more likely when classifiers are less accurate or more biased, or when effect sizes are small.

We argue that current common practices to address such limitations are insufficient. These practices assert validity by reporting classifier performance on manually annotated data quantified as metrics including accuracy, precision, recall, or the F1 score (Baden et al., 2022; Hase et al., 2022; Song et al., 2020). These steps promote confidence in results by making misclassification transparent, but our example indicates that high predictiveness may not protect researchers from biases flowing downstream into statistical inferences. Instead of practicing transparancy and hoping not to be mislead by misclassification bias, researchers can and should use validation data to correct misclassification bias.

These claims may surprise because of the wide-spread misconception that misclassification causes only conservative bias (i.e., bias towards null effects). This is believed because it is true for bivariate least squares regression when misclassifications are

nonsystematic (Carroll et al., 2006; Loken & Gelman, 2017; van Smeden et al., 2020).[2] 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, as shown in our example, misclassification bias can be anti-conservative (Carroll et al., 2006; Loken & Gelman, 2017; van Smeden et al., 2020). In regression models with more than one independent variable, or in nonlinear models, such as the logistic regression we used in our example, even nonsystematic misclassification can cause bias away from 0. Second, systematic misclassification can bias inference in any direction.

ACs designed in one context and applied in another are likely to commit systematic misclassification. For example, the Perspective API used to classify toxic content was developed for social media comments but performs much worse when applied to news data (Hede et al., 2021). Systematic misclassification may also arise when an AC used for measurement shapes behavior in a sociotechnical system under study. As examples, the Perspective API is used for online forum moderation (Hede et al., 2021), as is the ORES API for Wikipedia moderators (TeBlunthuis et al., 2021). Misclassifications from such classifiers can be systematic because they have causal effects on outcomes related to moderation.

If ACs become standard measurement devices, for instance Google's Perspective API for measuring toxicity (see critically Hosseini et al., 2017) or Botometer for classifying

---

[2] 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). Non-classical measurement error in an independent variable can be "differential" if it is not conditionally independent of the dependent variable given the other independent variables. Measurement error in an independent variable can be nondifferential and not classical. This is called Berkson and has the form $X = W + \xi$. 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 as predictions would then have lower variance than the training data. Following prior work, we thus do not consider Berkson errors (Fong & Tyler, 2021; Zhang, 2021). We call measurement error in the dependent variable *systematic* when it is correlated with an independent variable.

social media bots (see critically Rauchfleisch & Kaiser, 2020), entire literatures may have systematic biases. Even if misclassification bias is usually conservative, it can slow progress in a research area. Consider how Scharkow and Bachl (2017) argue that media's "minimal effects" on political opinions and behavior in linkage studies may be an artifact of measurement errors both in manual content analyses and self-reported media use in surveys. Conversely, if researchers selectively report statistically significant hypothesis tests, misclassification can introduce an upward bias in the magnitude of reported effect sizes and contribute to a replication crisis (Loken & Gelman, 2017).

Quantifying the Problem: Error Correction Methods in SML-based Text Classification

To understand how social scientists, including communication scholars, engage with the problem of misclassification in automated content analysis, we conducted a systematic literature review of studies using supervised machine learning (SML) for text classification (see Appendix B in our Supplement for details).[3] Our sample consists of studies identified by similar reviews on automated content analysis (Baden et al., 2022; Hase et al., 2022; Jünger et al., 2022; Song et al., 2020). Our goal 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. Studies used SML-based text classification for purposes such as to measure frames (Opperhuizen et al., 2019) or topics (Vermeer et al., 2020). They often employed SML-based ACs to create dichotomous

[3] Automated content analysis includes a range of 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; Hase, 2023). While we focus on SML, our arguments extend to other approaches such as dictionary-based classification and even beyond the specific context of text classification.

(50%) or other categorical variables (23%).[4] Of these empirical studies, many used SML-based ACs as independent variables (44%) or dependent variables (40%) in multivariate analyses, and 90% reported univariate statistics such as proportions.

Overall, our review reveals a *lack of transparency when reporting SML-based text classification*, similar to that previously reported (Reiss et al., 2022): A large share of studies do not report important methodological decisions related to sampling and sizes of training and test sets (see Appendix B). 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 misclassification bias nearly impossible. Most important, our review finds that *studies almost never reflected upon nor corrected misclassification bias*. According to our review, 85% of studies reported metrics such as recall or precision, but only 19% of studies explicitly stated that an AC misclassified texts which may introduce measurement error. Only a single article reported using error correction methods. To address the clear need for methods for understanding misclassification bias and correcting it, we now introduce and discuss existing methods to do so.

Addressing the Problem: Existing Approaches for Correcting Misclassification

Statisticians have extensively studied measurement error (including misclassification), the problems it causes for statistical inference, and methods for correcting these problems (see Carroll et al., 2006; Fuller, 1987). We narrow our focus to three existing methods recently proposed for dealing with misclassification bias in the context of automated content analysis: Fong and Tyler (2021)'s GMM calibration method, multiple imputation (Blackwell et al., 2017), and Zhang (2021)'s pseudo-likelihood model.[5]

---

[4] Metric variables were created in 35% of studies, mostly via the non-parametric method by Hopkins and King (2010).

[5] Statisticians have studied other methods including simulation extrapolation, Bayesian estimation, and score function methods. As we argue in Appendix C, these error correction methods are not advantageous

In the interest of clarity, we introduce some notation. Say we want to estimate a regression model $Y = B_0 + B_1X + B_2Z + \varepsilon$ where X is an independent variable for which a small sample of manually annotated data X^* and automated classifications W are observed. Fully observed are Z, a second independent variable and Y, the dependent variable. To illustrate, in our first real-world example, X is toxicity, X^* are the civil comment annotations, W are the Perspective API's toxicity classification, Z are likes, and Y is identity disclosure.

Say the sample of annotated data X^* is too small to convincingly test a hypothesis, but collecting additional annotations is too expensive. In contrast, an AC can make classifications W for the entire dataset but introduces misclassification bias. How can we correct this bias in an automated content analysis?

Regression calibration uses observable variables, including automated classifications W and other variables measured without error Z, to approximate the true value of X (Carroll et al., 2006). Fong and Tyler (2021) propose a regression calibration procedure designed for SML that we refer to as *GMM calibration* or GMM.[6] For their calibration model, Fong and Tyler (2021) use 2-stage least squares (2SLS). They regress the observed variables Z and AC predictions W onto the manually annotated data and then use the resulting model to approximate X as $\hat{X}$. They then use the generalized method of moments (gmm) to combine estimates based on the approximated independent variable $\hat{X}$ and estimates based on the manually annotated data X^*. This method makes efficient use of manually annotated data and provides an asymptotic theory for deriving confidence intervals. The GMM approach does not make strong assumptions about the distribution of the outcome Y, but can be invalidated by systematic misclassification (Fong & Tyler, 2021). GMM, like other regression calibration techniques, is not designed to correct for

---

when manually annotated data is available, as is often the case with ACs.

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

misclassification in the outcome.

Multiple imputation (MI) treats misclassification as a missing data problem. It understands the true value of X to be observed in manually annotated data X^* and missing otherwise (Blackwell et al., 2017). 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 X conditional on other variables $\{W, Y, Z\}$, then runs a statistical analysis on each of these sampled datasets and pools the results of each of these analyses (Blackwell et al., 2017). Note that Y is included among the imputing variables, giving the MI approach the potential to address differential error. Blackwell et al. (2017) claim that the MI method is relatively robust when it comes to small violations of the assumption of nondifferential error. Moreover, in theory, the MI approach can be used for correcting misclassifications both in independent and dependent variables.

"Pseudo-likelihood" methods (PL)—even if not always explicitly labeled this way—are another approach for correcting misclassification bias. 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 independent variable and the AC's false positive and false negative rates in the case of a mismeasured dependent variable. 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 is that this method only requires summary quantities derived from manually annotated data, for instance via a confusion matrix.

Proposing a Likelihood Modeling Approach to Correct Misclassification

We now elaborate on a new *Maximum Likelihood Method* (MLE) we propose for correcting misclassification bias. Our method tailors Carroll et al. (2006)'s presentation of the general statistical theory of likelihood modeling for measurement error correction to

context of automated content analysis.[7] The MLE approach deals with misclassification bias by maximizing a likelihood that correctly specifies an *error model* of the probability of the automated 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 possible values of a variable to the likelihood, weighted by the likelihood of the error model.

MLE methods have four advantages in the context of ACs. First, they are general in that they 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 manually annotated data to make precise estimates. Third, the MLE approach is applicable both for correcting for misclassification in a dependent and an independent variable. Fourth, and most important, this approach is effective when misclassification is systematic.

When an Automated Classifier Predicts an Independent Variable

In general, if we want to estimate a model $P(Y|\Theta_Y, X, Z)$ for Y given X and Z with parameters Θ_Y, we can use AC classifications W predicting X to gain statistical power without introducing misclassification bias by maximizing $(\mathcal{L}(\Theta|Y, W))$, the likelihood of the parameters $\Theta = \{\Theta_Y, \Theta_W, \Theta_X\}$ in a joint model of Y and the error model of W (Carroll et al., 2006). The joint probability of Y and W, can be factored into the product of three terms: $P(Y|X, Z, \Theta_Y)$, the model we want to estimate, $P(W|X, Y, \Theta_W)$, a model for W having parameters Θ_W, and $P(X|Z, \Theta_X)$, a model for X having parameters Θ_X. Calculating these three conditional probabilities is sufficient to calculate the joint probability of the dependent variable and automated classifications and thereby obtain a

[7] In particular see Chapter 8 (especially example 8.4) and Chapter 15. (especially 15.4.2).

consistent estimate despite misclassification. $P(W|X, Y, \Theta_W)$ is called the *error model* and $P(X|Z, \Theta_X)$ is called the *exposure model* Carroll et al., 2006.

To illustrate, the regression model $Y = B_0 + B_1X + B_2Z + \varepsilon$, predicts the discrete independent variable X. 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 $\Theta = \{\Theta_Y, \Theta_W, \Theta_X\} = \{\{B_0, B_1, B_2\}, \{\alpha_0, \alpha_1, \alpha_2\}, \{\gamma_0, \gamma_1\}\}$.

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

$$P(Y_i|X_i, Z_i, \Theta_Y) = \phi(B_0 + B_1X_i + B_2Z_i) \tag{2}$$

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

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

where ϕ is the normal probability distribution function. Note that Equation 1 models differential error (i.e., Y is not independent of W conditional on X and Z) via 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 $P(W|X, Y, Z, \Theta_W)$ and $P(X|Z, \Theta_X)$ (Williams & Barber, 1998).

When an Automated Classifier Predicts a Dependent Variable

We now turn to the case when an AC makes classifications W that predict a discrete dependent variable Y. In our second real-data example, W is the Perspective API's toxicity classifications and Y is the true value of toxicity. This case is simpler than

the case above where an AC is used to measure an independent variable 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 to 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 W, the dependence of W on X and Z can be omitted from equations 5 and 7. Additional details on the likelihood modeling approach available in Appendix D of the Supplement.

Evaluating Misclassification Models: Monte-Carlo Simulations

We now present four Monte Carlo simulations (*Simulations 1a*, *1b*, *2a*, and *2b*) with which we evaluate existing methods (GMM, MI, PL) and our approach (MLE) for correcting misclassification bias.

Monte Carlo simulations are a 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). They are defined by a data generating process from which datasets are repeatedly sampled. Repeating an analyses for each of these datasets provides an empirical distribution of results the analysis would obtain over study replications. Monte-carlo simulation affords exploration of finite-sample performance, robustness to assumption violations, comparison across several methods, and ease of interpretability (Mooney, 1997).

Parameters of the Monte Carlo Simulations

In our simulations, we tested four error correction methods: *GMM calibration* (GMM) (Fong & Tyler, 2021), *multiple imputation* (MI) (Blackwell et al., 2017), *Zhang's pseudo-likelihood model* (PL) (Zhang, 2021), and our *likelihood modeling* approach (MLE). We use the `predictionError` R package (Fong & Tyler, 2021) for the GMM method, the `Amelia` R package for the MI approach, and our own implementation of Zhang (2021)'s PL approach in R. We develop our MLE approach in the R package `misclassificationmodels`. For PL and MLE, we quantify uncertainty using the fisher information quadratic approximation.

In addition, we compare these error correction methods to two common approaches in communication science: the *feasible* estimator (i.e., conventional content analysis that uses only manually annotated data and not ACs) and the *naïve* estimator (i.e., using AC-based classifications W as stand-ins for X, thereby ignoring misclassifications). According to our systematic review, the *naïve* approach reflects standard practice in studies employing SML for text classification.

We evaluate each of the six analytical approaches 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 data), and *uncertainty quantification* (how well the 95% confidence intervals approximate the range including 95% of parameter estimates across simulations). To evaluate efficiency, we repeat each simulation with different amounts of total observations, i.e., unlabeled data to be classified by an AC (ranging from 1000 to 10000 observations), and manually annotated observations (ranging from 100 to 400 observations). Since our review indicated that ACs are most often used to create binary variables, we restrict our simulations to misclassifications related to a binary (in-)dependent variable.

Four Prototypical Scenarios for our Monte Carlo Simulations

We simulate regression models with two independent variables (X and Z). This sufficiently constrains our study's scope but the scenario 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 independent variables, the dependent variable Y, and automated classifications W. This structure determines if systematic misclassifications in an independent variable cause differential error and if systematic misclassifications in a dependent variable should be modeled (Carroll et al., 2006). In Figure 2, we illustrate our scenarios via Bayesian networks representing the conditional dependence structure of variables (Pearl, 1986): We first simulate two cases where an AC measures an independent variable without (*Simulation 1a*) and with differential error (*Simulation 1b*). Then, we simulate using an AC to measure the dependent variable, either one with misclassifications that are uncorrelated (*Simulation 2a*) or correlated with an independent variable (*Simulation 2b*). GMM is not designed to correct misclassifications in dependent variables, so we omit this method in *Simulations 2a* and *2b*.

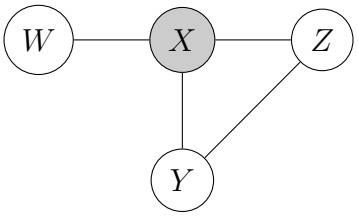
*Misclassification in an Independent Variable (*Simulations 1a *and* 1b*)*

We first consider studies with the goal of testing hypotheses about the coefficients B_1 and B_2 in a least squares regression:

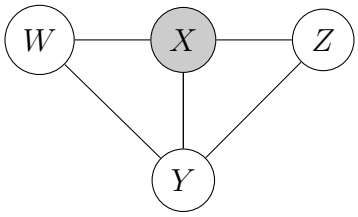
$$Y = B_0 + B_1 X + B_2 Z + \varepsilon \tag{8}$$

In our first real-data example, Y was a discrete variable-whether a comment self-disclosed a racial or ethnic identity, X was if a comment was toxic, and Z was the number of likes. In this simulated 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.

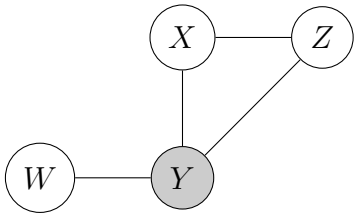
Both simulations have a normally distributed dependent variable Y and two binary



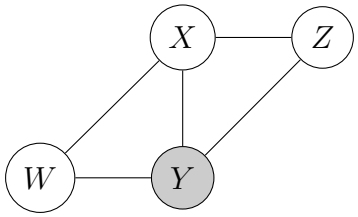
(a) *In* Simulation 1a*, classifications W are conditionally independent of Y so a model using W as a proxy for X has non-differential error.*



(b) *In* Simulation 1b*, the edge from W to Y signifies that the automatic classifications W are not conditionally independent of Y given X, indicating differential error.*



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



(d) *In* Simulation 2b*, the edge connecting W and X signifies that the predictions W are not conditionally independent of X given Y, indicating systematic misclassification.*

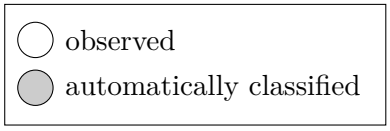


Figure 2

Bayesnet networks representing the conditional independence structure of our simulations.

independent variables X and Z, which are balanced ($P(X) = P(Z) = 0.5$) and correlated (Pearson's $\rho = -0.12$). To represent a study design where an AC is needed to obtain sufficient statistical power, Z and X can explain only 10% of the variance in Y.

In *Simulation 1a* (Figure 2a), we simulate an AC with 72% accuracy.[8] This reflects a situation where X may be difficult to predict, but the AC, represented as a logistic regression model having linear predictor W^*, provides a useful signal. We simulate nondifferential misclassification because $W = X + \xi$, ξ is normally distributed with mean 0, and ξ and W are conditionally independent of Y given X and Z.

In our first real-data example, the Perspective API predicted comment toxicity, which was an independent variable of a regression model in which racial/ethnic identity disclosure was the dependent variable. The API disproportionately misclassified as toxic comments disclosing such identities which toxic which resulted in differential misclassification.

In *Simulation 1b* (Figure 2b), we test how error correction methods can handle differential error by making AC predictions similarly depend on the dependent variable Y. This simulated AC has 74% accuracy and 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.

Measurement Error in a Dependent Variable (Simulation 2a and 2b)

We then simulate using an AC to measure the dependent variable Y, a binary independent variable X, and a continuous independent variable Z. The goal is to estimate B_1 and B_2 in the following logistic regression model:

---

[8] Classifier accuracy varies between our simulations because it is difficult to jointly specify classifier accuracy and the required correlations among variables and due to random variation between simulation runs. We report the median accuracy over simulation runs.

$$P(Y) = \frac{1}{1 + e^{-(B_0 + B_1 X + B_2 Z)}} \tag{9}$$

In our second real-data example, Y is if a comment contains toxicity, X is if the comment discloses racial or ethnic identity, and Z is the number of times the comment was "liked".

In *Simulation 2a* (see Figure 2c) and *Simulation 2b* (see Figure 2d) X and Z are, again, balanced ($P(X) = P(Z) = 0.5$) and correlated (Pearson's $\rho = -0.12$). 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. We therefore choose, somewhat arbitrarily, $B_X = 0.7$ and $B_Z = -0.7$. We again simulate ACs with moderate predictive performance. The AC in *Simulation 2a* is 72% accurate and the AC in *Simulation 2b* is 71% accurate. In *Simulation 2a*, the misclassifications are nonsystematic as ξ has mean 0 and is independent of X and Z. However, in *Simulation 2b* the misclassifications ξ are systematic and correlated with Z (Pearson's $\rho = -0.18$).

Simulation Results

For each method, we visualize the consistency, efficiency, and the accuracy of uncertainty quantification of estimates across prototypical scenarios. For example, Figure 3 visualizes results for *Simulation 1a*. Each subplot shows a simulation with a given total sample size (No. observations) and a given sample of manually annotated observations (No. manually annotated observations). To assess a method's consistency, we locate the expected value of the point estimate across simulations with the center of the black circle. As an example, see the leftmost column in the bottom-left subplot of Figure 3. For the naïve estimator, the circle is far below the dashed line indicating the true value of B_X. Here, ignoring misclassification causes bias toward 0 and the estimator is inconsistent. To assess a method's efficiency, we mark the region in which point estimate falls in 95% of the simulations with black lines. The black lines in the bottom-left subplot of Figure 3 for example show that the feasible estimator, which uses only manually annotated data, is consistent but less precise than estimates from error correction methods. To assess each

method's uncertainty quantification, compare the gray lines, which show the expected value of a method's approximate 95% confidence intervals across simulations, to the neighboring black lines. The *PL* column in the bottom-left subplot of Figure 3 for instance shows that the method's 95% confidence interval is biased towards 0 when the number of manually annotated observations is smaller. This is to be expected because the PL estimator does not account for uncertainty in misclassification probabilities estimated using the sample of manually annotated observations.

Simulation 1a: Nonsystematic Misclassification of an Independent Variable

Figure 3 illustrates *Simulation 1a*. Here, the naïve estimator is severely biased in its estimation of B_X. Fortunately, error correction methods (GMM, MI, MLE) produce consistent estimates and acceptably accurate confidence intervals. Notably, the PL method is inconsistent and considerable bias remains when the sample of annotations is much smaller than the entire dataset. This is likely due to $P(X = x)$ missing from the PL estimation.[9] Figure 3 also shows that MLE and GMM estimates become more precise in larger datasets. This is is less pronounced for MI estimates, indicating that GMM and MLE use automated classifications more efficiently than MI.

In brief, when misclassifications cause nondifferential error, MLE and GMM are effective, efficient, and provide accurate uncertainty quantification. They complement each other due to different assumptions: MLE depends on correctly specifying the likelihood but its robustness to incorrect specifications is difficult to analyze (Carroll et al., 2006). The GMM approach depends on the exclusion restriction instead of distributional assumptions (Fong & Tyler, 2021). MLE's advantage over GMM come from the relative ease with which it can be extended to for instance generalized linear models (GLMs) or generalized additive models (GAMs). In cases similar to *Simulation 1a*, we therefore recommend both the GMM and an appropriately specified MLE approach to correct for misclassification.

---

[9] Compare Equation D4 in Appendix D to Equations 24-28 from Zhang (2021).

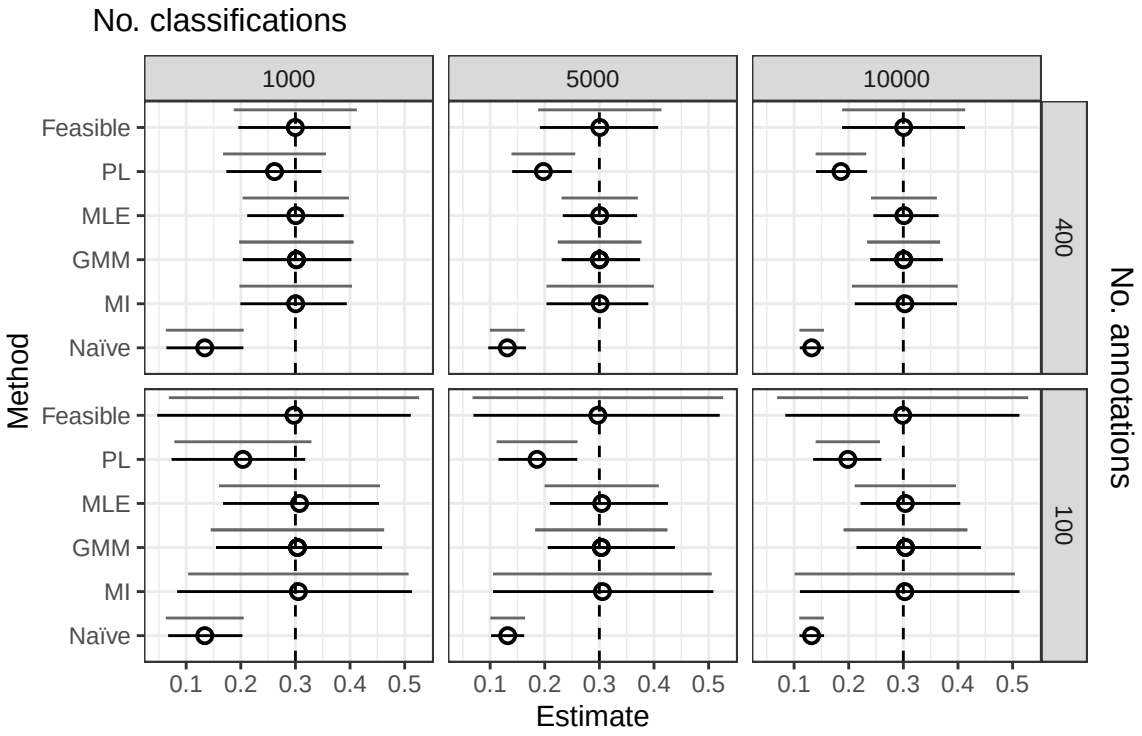


Figure 3

Simulation 1a: Nonsystematic misclassification of an independent variable. Error correction methods, except for PL, obtain precise and accurate estimates given sufficient manually annotated data.

Simulation 1b: Systematic Misclassification of an Independent Variable

Figure 4 illustrates *Simulation 1b*. Here, systematic misclassification gives rise to differential error and creates more extreme misclassification bias that is more difficult to correct. As Figure 4 shows, the naïve estimator is opposite in sign to the true parameter. Of the four methods we test, only the MLE and the MI approach provide consistent estimates. This is expected because they use Y to adjust for misclassifications. The bottom row of Figure 4 shows how the precision of the MI and MLE estimates increase with additional observations. As in *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 F2 in Appendix F 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. Inspecting results from methods that do not correct for differential error is useful for understanding their limitations. When few annotations of X are observed, GMM is nearly as bad as the naïve estimator. PL is also visibly biased. Both improve when a greater proportion of the data is labeled since they combine AC-based estimates with the feasible estimator.

In sum, our simulations suggest that the MLE approach is superior in conditions of differential error. Although estimations by the MI approach are consistent, the method's practicality is limited by its inefficiency.

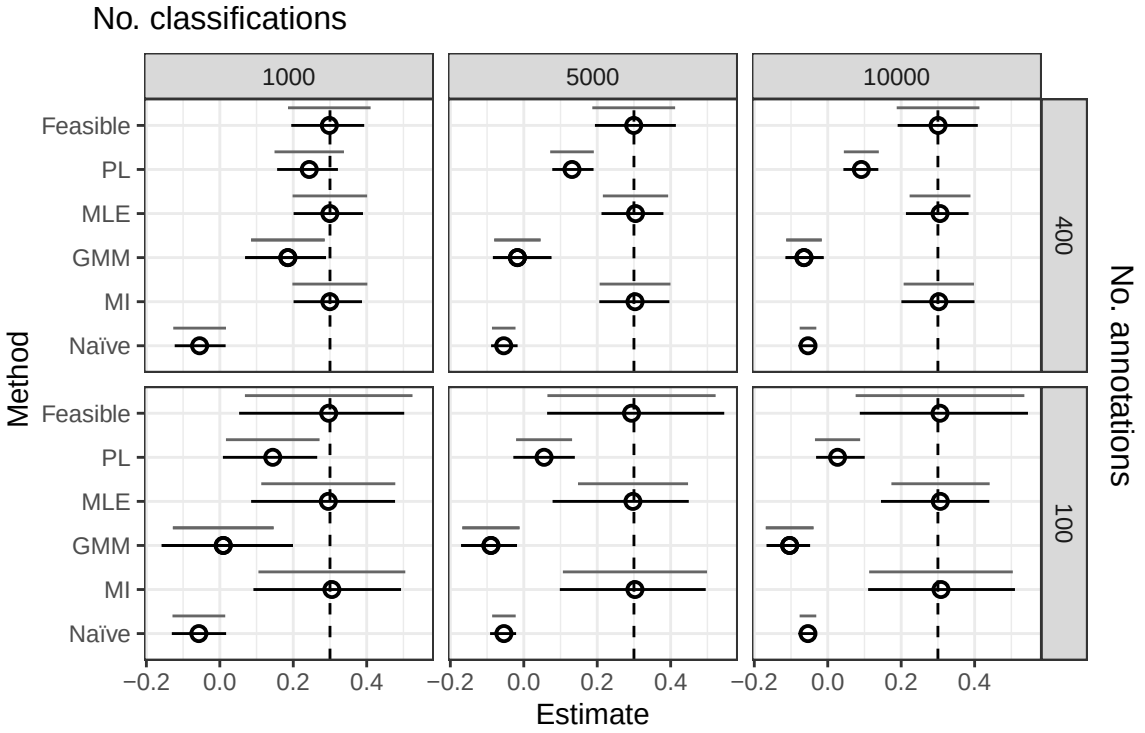


Figure 4

Simulation 1b: Systematic misclassification of an independent variable. Only the the MLE approach obtains consistent estimates of B_X.

Simulation 2a: Nonsystematic Misclassification of a Dependent Variable

Figure 5 illustrates *Simulation 2a*: nonsystematic misclassification of a dependent variable. This also introduces bias as evidenced by the naïve estimator's inaccuracy. Our MLE method is able to correct this error and provide consistent estimates. Surprisingly, the MI estimator is inconsistent and does not improve with more human-labeled data. The PL approach is also inconsistent, especially when only few of all observations are annotated manually. It is closer to recovering the true parameter than the MI or the naïve estimator, but provides only modest improvements in precision compared to the feasible estimator. It is clear that the precision of the MLE estimator improves with more observations data to a greater extent than the PL estimator. When the amount of human-labled data is low, inaccuracies in the 95% confidence intervals of both the MLE and PL become visible due to the poor finite-sample properties of the quadradic approximation for standard errors.

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

Simulation 2b: Systematic Misclassification of a Dependent Variable

In *Simulation 2b*, misclassifiations of the dependent variable Y are correlated with an independent variable X. As shown in Figure 6, this causes dramatic bias in the naïve estimator. Similar to *Simulation 2a*, MI is inconsistent. PL is also inconsistent because it does not account for Y when correcting for misclassifications. 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 F4 in Appendix F shows that the precision of estimates for the coefficient for X improves with additional data to a greater extent. As such, 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 dependent variable are correlated with an independent variable.

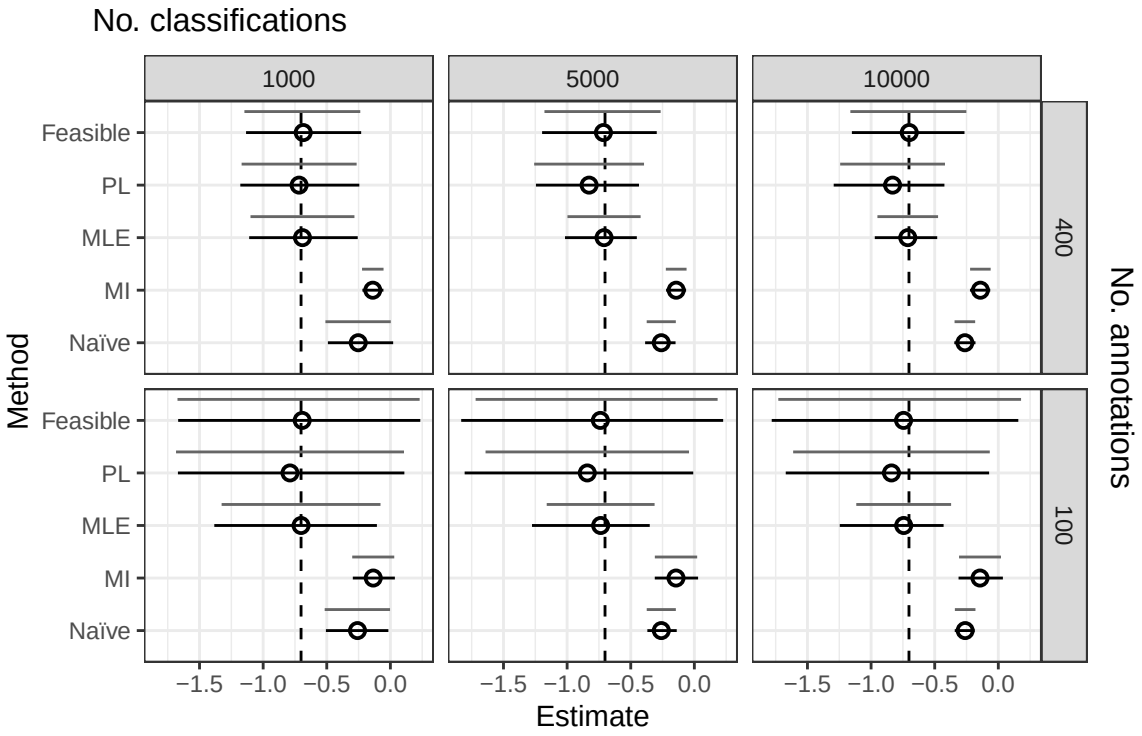


Figure 5

Simulation 2a: Nonsystematic misclassification of a dependent variable. Only the MLE approach obtains consistent estimates.

Transparency about Misclassification Is Not Enough—We Have To Fix It! Recommendations for Automated Content Analysis

"Validate, Validate, Validate" (Grimmer & Stewart, 2013, p. 269) 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 validity, for instance by making misclassification transparent. 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 some external ground truth sufficient to claim that results are valid? Or is there something else we can do and should do?

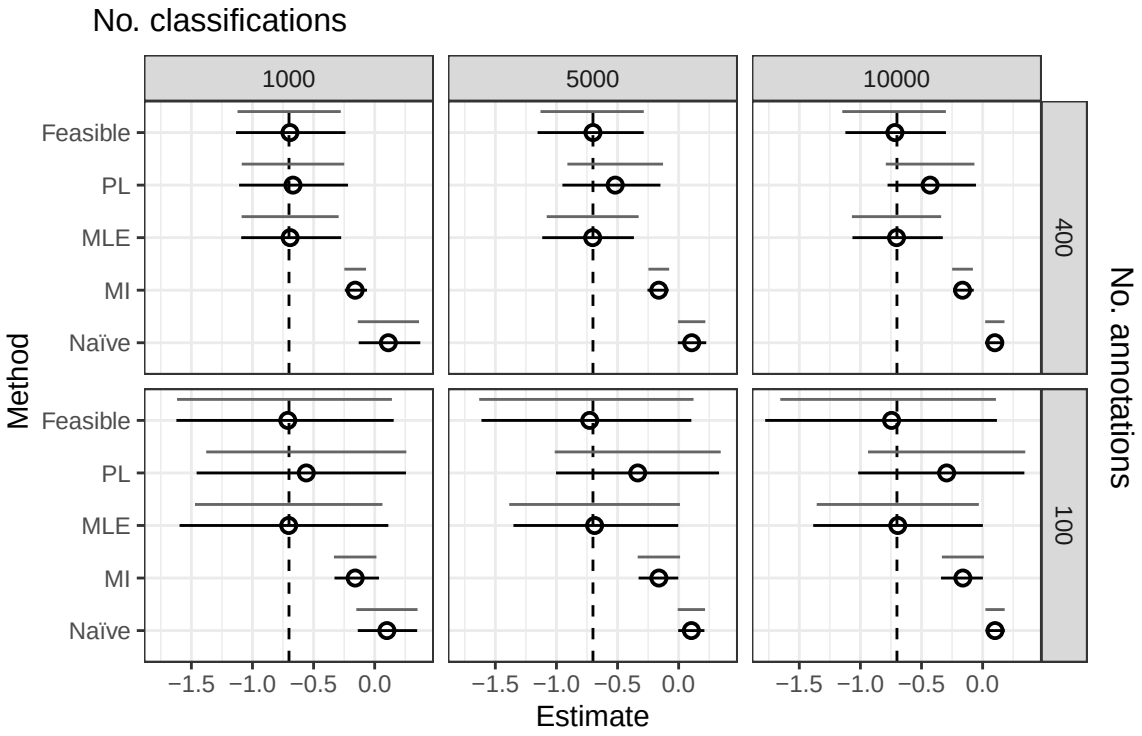


Figure 6

Simulation 2b: Systematic misclassification of a dependent variable. Only the MLE approach obtains consistent estimates.

We think there is: Using statistical methods to not only quantify but also correct for misclassification. Our study provides several recommendations in this regard, with an overview of recommendations provided in Figure 7.

Step 1: Attempt Manual Content Analysis

Manual content annotation is often done *post facto*, for instance to calculate predictiveness of an already existing AC such as Google's Perspective classifier. We propose to instead use manually annotated data *ante facto*, i.e. before building or validating an AC. Practically speaking, the main reason to use an AC is feasibility: to avoid the costs of manual coding a large dataset. One may for example need a large dataset to study an effect one assumes to be small. Manually labeling such a dataset is expensive. Often, ACs are seen as a cost-saving procedure without consideration of the

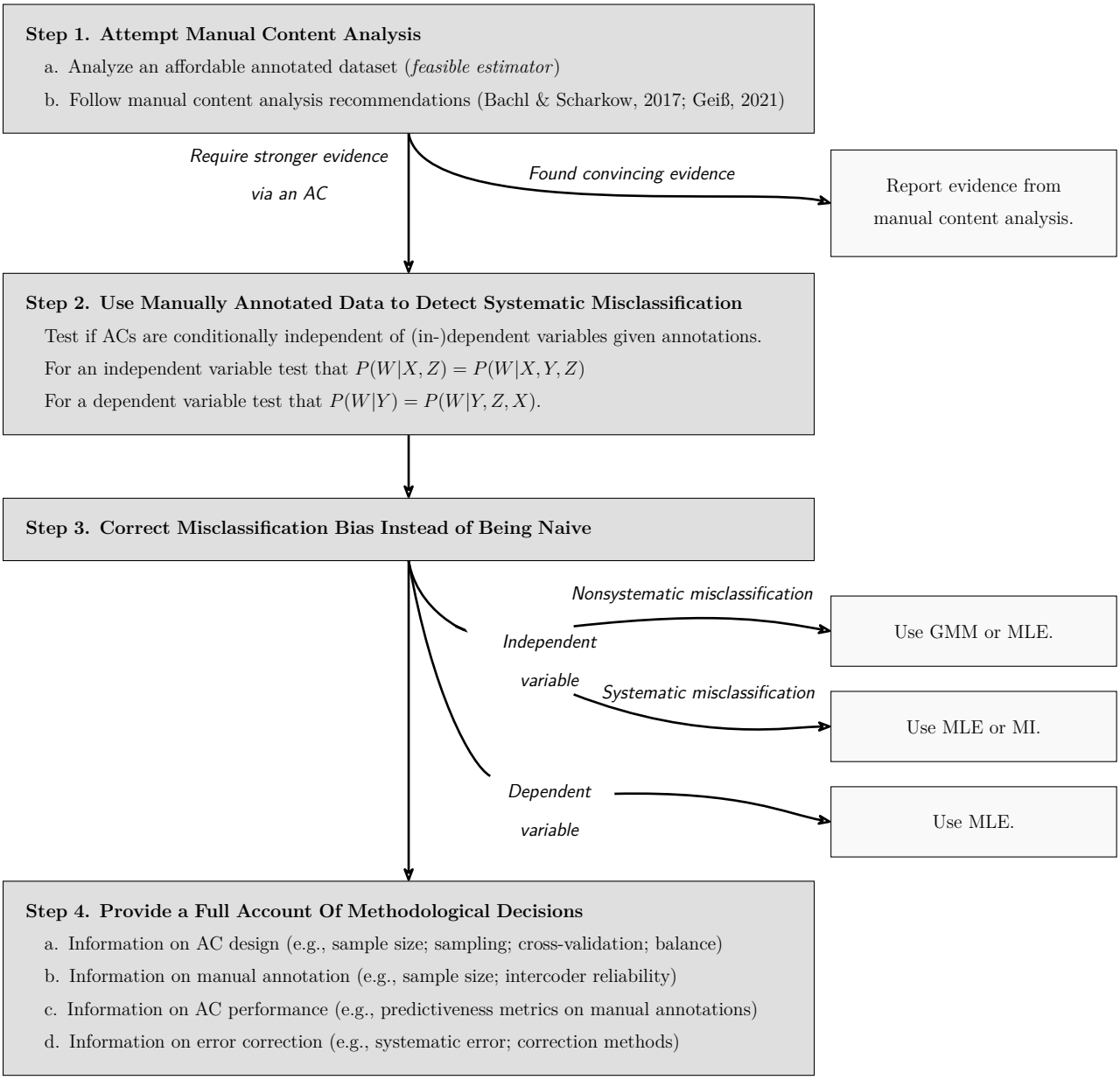


Figure 7

Recommendations for Automated Content Analysis Study Design

threats to validity posed by misclassification. Moreover, validating an existing AC or building a new AC is also expensive, for instance due to costs of computational resources or manual annotation of (perhaps smaller) test and training datasets.

We therefore caution researchers against preferring automated over manual content

analysis unless doing so is necessary to obtain useful evidence. We agree with Baden et al. (2022) who argue that "social science researchers may be well-advised to eschew the promises of computational tools and invest instead into carefully researcher-controlled, limited-scale manual studies" (p. 11). In particular, we recommend to use manually annotated data *ante facto*: Researchers should begin by statistical modeling human-annotated data so to discern if an AC is necessary. In our simulations, the feasible estimator is less precise but consistent in all cases. So if fortune shines and this estimate sufficiently answers one's research question, manual coding is sufficient. Here, scholars should rely on existing recommendations for descriptive and inferential statistics in the context of manual content analysis (Bachl & Scharkow, 2017; Geiß, 2021). If the feasible estimator however fails to provide convincing evidence, for example by not rejecting the null, manually annotated data is not wasted. It can be reused to build an AC or correct misclassification bias.

Step 2: Use Manually Annotated Data to Detect Systematic Misclassification

As demonstrated in our simulations, knowing whether an AC makes systematic misclassifications is important: It determines which correction methods can work. Fortunately, manually annotated data can be used to detect systematic misclassification. 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. More generally, one can test if the data's conditional independence structures can represented by Figures 2a or 2c. This can be done, for example, via likelihood ratio tests of $P(W|X,Z) = P(W|X,Y,Z)$ (if an AC measures an independent variable X) or of $P(W|Y) = P(W|Y,Z,X)$ (if an AC measures a dependent variable Y) or by visual inspection of plots of relating misclassifications to other variables (Carroll et al., 2006). We strongly recommend using such methods to test for differential error and to design an appropriate correction.

Step 3: Correct for Misclassification Bias 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. As shown in Appendix **??**, this method requires specifying a valid error model to obtain consistent estimates. This may not be too difficul in practice because if one can assume the primary model for Y, this implies that an error model for W that includes all observed variables is sufficient. Still, on should take care correctly model nonlinearities and interactions. Our **misclassificationmodels** R package provides reasonable default error models and a user-friendly interface to facilitate adoption of our MLE method (see Appendix E).

When feasible, we recommend comparing the MLE approach to another error correction method. Consistency between two correction methods shows that results are robust independent of the correction method. If the AC is used to predict an independent variable, GMM is a good choice if error is nondifferential. Otherwise, MI can be considered. Unfortunantly, if the AC is used to predict a dependent variable, our simulations do not support a strong suggestion for a second method. PL might be useful reasonable choice with enough manually annotated data and non-differential error. This range of viable choices in error correction methods also motivates our next recommendation.

Step 4: Provide a Full Account of Methodological Decisions

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, number of annotators, intercoder values, size of data annotated

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.

Conclusion and Limitations

In this study, we discuss the problem of misclassification in automated content analysis which may introduce misclassification bias in statistical models. We believe this is a topic that has not attracted enough attention within communication science (but see Bachl & Scharkow, 2017) and even in the broader computational social science community. After illustrating biases emerging from automated classifiers such as the Perspective API, we quantify how aware researchers are of the issue of misclassification. In a systematic review of studies using SML-based text classification, we show that scholars rarely acknowledge this problem and almost never address it. We therefore discuss a range of statistical methods that use manually annotated 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 situations involving differential error. Based on these results, we provide four recommendations for the future of automated content analysis: Researchers should (1) attempt manual content analyssis before building or validating ACs to see whether human-labeled data is sufficient, (2) use manually annotated data to test for systematic misclassification and choose appropiate error correction methods, (3) correct for misclassifications via error correction methods, and (4) be transparent about the methodological decisions involved in AC-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 annotators do not make errors, especially systematic ones. This assumption can be reasonable if intercoder reliability is very high but, as with ACs, this may not always be the case. Thus, it may be important to account for measurement error by human coders (Bachl & Scharkow, 2017) and by automated 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 automated 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 all possible factors that may influence the performance and robustness of error correction methods including classifier accuracy, heteroskedasticity, and violations of distributional assumptions. We are working to investigate such factors, as shown in Appendix F, by extending our simulations. Third, we simulated datasets with balanced variables, but classifiers are often used to measure rare occurrences. Imbalanced covariates will require greater sample sizes of validation data to correct for misclassification. In such cases, validation data may be collected more efficiently using approaches that provide balanced, but unrepresentative samples. However, non-representative sampling requires correction methods to account for the probability that a data point will be sampled.

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.

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness in Machine Learning*. fairmlbook.org.

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. (2017). A Unified Approach to Measurement Error and Missing Data: Details and Extensions. *Sociological Methods & Research*, *46*(3), 342–369.

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.

Burscher, B., Vliegenthart, R., & De Vreese, C. H. (2015). Using Supervised Machine Learning to Code Policy Issues: Can Classifiers Generalize across Contexts? *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 122–131.

Burscher, B., Odijk, D., Vliegenthart, R., de Rijke, M., & de Vreese, C. H. (2014). Teaching the Computer to Code Frames in News: Comparing Two Supervised Machine Learning Approaches to Frame Analysis. *Communication Methods and Measures*, *8*(3), 190–206.

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

cjadams, Daniel Borkan, inversion, Jeffery Sorensen, Lucas Dixon, Lucy Vasserman, & nithum. (2019). Jigsaw Unintended Bias in Toxicity Classification.

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.

Grimmer, J., Roberts, M. E., & Stewart, B. M. (2021). Machine Learning for Social Science: An Agnostic Approach. *Annual Review of Political Science*, *24*(1), 395–419.

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. (2023). Automated Content Analysis. In F. Oehmer-Pedrazzi, S. H. Kessler, E. Humprecht, K. Sommer, & L. Castro (Eds.), *Standardisierte Inhaltsanalyse in der Kommunikationswissenschaft – Standardized Content Analysis in Communication Research* (pp. 23–36). Springer Fachmedien Wiesbaden.

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). From Toxicity in Online Comments to Incivility in American News: Proceed with Caution. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2620–2630.

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.

Hopp, T., & Vargo, C. J. (2019). Social Capital as an Inhibitor of Online Political Incivility: An Analysis of Behavioral Patterns Among Politically Active Facebook Users. *International Journal of Communication*, *13*(0), 21.

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.

Kim, J. W., Guess, A., Nyhan, B., & Reifler, J. (2021). The Distorting Prism of Social Media: How Self-Selection and Exposure to Incivility Fuel Online Comment Toxicity. *Journal of Communication*, *71*(6), 922–946.

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

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*. Annual Conference of the Methods Section of the German Communication Section, Munich.

Scharkow, M. (2013). Thematic content analysis using supervised machine learning: An empirical evaluation using German online news. *Quality & Quantity*, *47*(2), 761–773.

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.

TeBlunthuis, N., Hill, B. M., & Halfaker, A. (2021). Effects of Algorithmic Flagging on Fairness: Quasi-experimental Evidence from Wikipedia. *Proceedings of the ACM on Human-Computer Interaction*, *5*, 56:1–56:27.

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.

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.

Votta, F., Noroozian, A., Dobber, T., Helberger, N., & de Vreese, C. (2023). Going Micro to Go Negative?: Targeting Toxicity using Facebook and Instagram Ads. *Computational Communication Research*, *5*(1), 1–50.

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.

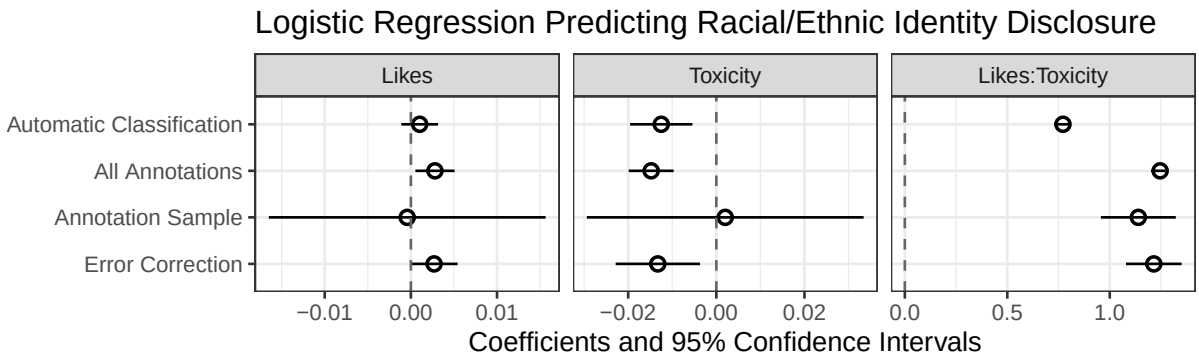
Zhang, H. (2021, May 29). *How Using Machine Learning Classification as a Variable in Regression Leads to Attenuation Bias and What to Do About It* (preprint). SocArXiv.

Appendix A

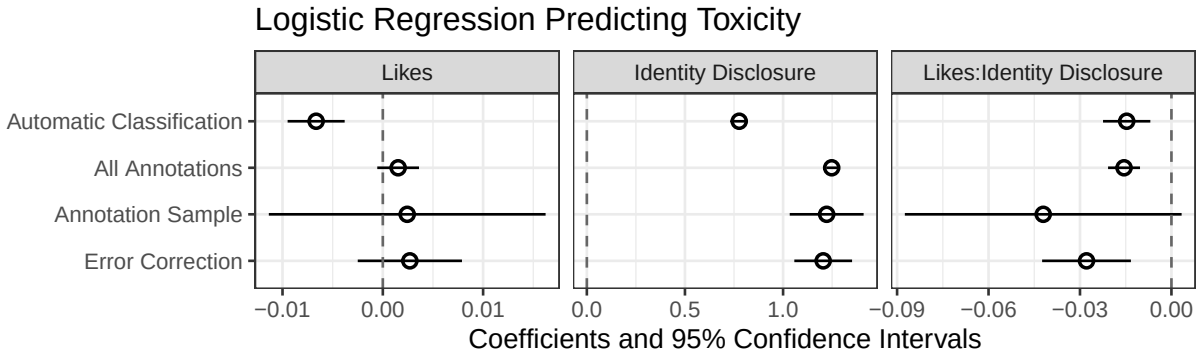
Perspective API Example

Our example relies on the publicly available Civil Comments dataset (cjadams et al., 2019). The dataset contains around 2 million comments collected from independent English-language news sites between 2015 and 2017. We rely on a subset of 448,000 comments which were manually annotated both for toxicity (*toxicity*) and disclosure of identity (*disclosure*) in a comment. The dataset also includes counts of likes each comment received (*number of likes*).

Each comment was labeled by up to ten manual annotators (although selected comments were labeled by even more annotators). Originally, the dataset represents *toxicity* and *disclosure* as proportions of annotators who labeled a comment as toxic or as disclosing aspects of personal identity including race and ethnicity. For our analysis, we converted these proportions into indicators of the majority view to transform both variables to a binary scale.



(a) Example 1*: Misclassification in an independent variable.*



(b) Example 2*: Misclassification in a dependent variable.*

Figure A1

Real-data example including correction using MLE.

Appendix B

Systematic Literature Review

To inform our simulations, we reviewed studies using SML for text classification.

Identification of Relevant Studies

Our sample was drawn from four recent reviews on the use of AC within the context of communication science and the social sciences more broadly (Baden et al., 2022; Hase et al., 2022; Jünger et al., 2022; Song et al., 2020). Authors of respective studies had either already published their data in an open-science approach or thankfully shared their data with us when contacted. From their reviews, we collected $N = 110$ studies that included some type of SML (for an overview, see Figure B1).

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 ($N = 10$, $\alpha = .89$), we excluded studies not fulfilling inclusion criteria, here studies not including any SML approach and studies only using SML for data cleaning, not data analysis—for instance to sort out topically irrelevant articles. Next, we removed studies focusing on methodologically advancing SML-based ACs since these studies often include far more robustness and validity tests than commonly employed in empirical settings. Subsequently, all relevant empirical studies ($N = 48$) were coded in further detail.

Manual Coding of Relevant Empirical Studies

For manual coding, we created a range of variables (for an overview, see Table B1). 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 authors 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), and *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), and *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), and *Accuracy of Classifier* (Open String),
- whether articles mentioned and/or corrected for misclassifications using the variables *Error Mentioned* (0 = No, 1 = Yes) and *Error Corrected* (0 = No, 1 = Yes).

Results

SML-based ACs were most often used to create dichotomous measurements (*Dichotomous*: 50%), followed by variables on a metric (*Metric*: 35%), categorical (*Categorical*: 23%), or ordinal scale (*Ordinal*: 10%). Almost all studies used SML-based classifications to report descriptive statistics on created variables (*Descriptive*: 90%). However, many also used these in downstream analyses, either as dependent variables (*Dependent*: 40%) or independent variables (*Independent*: 44%) in statistical models.

Only slightly more than half of all studies included information on the size of training or test sets (*Size Training Data*: 67%, *Size Test Data*: 52%). Even fewer included information on the size of manually annotated data for intercoder testing (*Size Data Intercoder Test*: 44%) or respective reliability values (*Intercoder Reliability*: 56%). Lastly, not all studies reported how well their classifier performed by using metrics such as precision, recall, or F1-scores (*Accuracy of Classifier*: 85%). Lastly, few studies exlicitly mentioned the issue of misclassification (*Error Mentioned*: 19%, with only a single study correcting for such (*Error Corrected*: 2%).

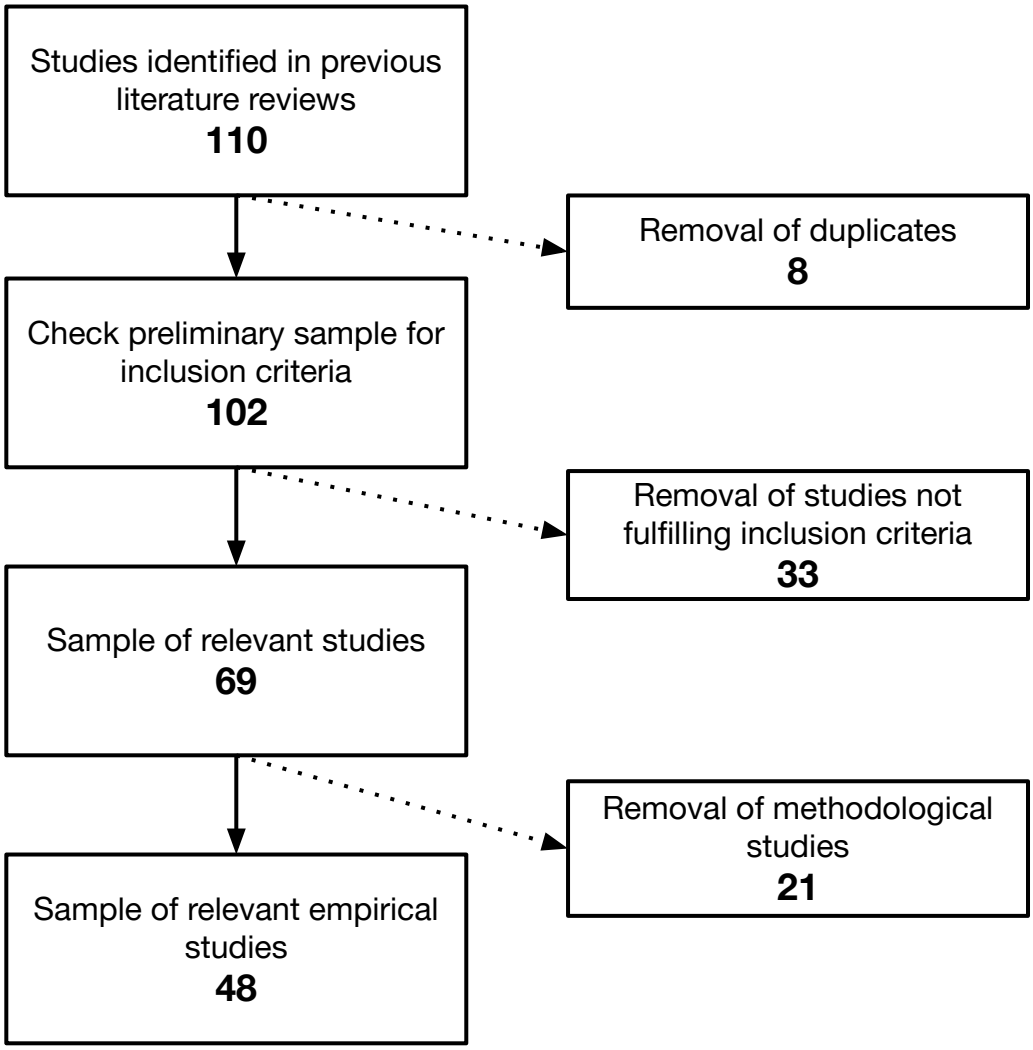


Figure B1

Identifying relevant studies for the literature review

Table B1

Variables Coded for Relevant Empirical Studies

Category	Variable	Krippendorf's α	% or M (SD)
Type of Journal	*Communication*	n.a.	67%
	Impact	n.a.	$M = 4$
Type of Variable	*Dichotomous*	0.86	50%
	Categorical	1	23%
	Ordinal	0.85	10%
	Metric	1	35%
Use of Variable	*Descriptive*	0.89	90%
	Independent	1	44%
	Dependent	1	40%
Information on Classifier	*Size Training Data*	0.95	67%
	Size Test Data	0.79	52%
	Size Data Intercoder Test	1	44%
	Intercoder Reliability	0.8	56%
	Accuracy of Classifier	0.77	85%
Measurement Error	*Error Mentioned*	1	19%
	Error Corrected	1	2%

Appendix C

Other Error Correction Methods

Statisticans have introduce a range of other error correction methods which we did not test in our simulations. Here, we shortly discuss three additional methods and explain why we did not include them in our simulations.

Simulation extrapolation (SIMEX) simulates 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 manually annotated data, but may be more complicated than necessary to correct measurement error from ACs when manually annotated data is available. Likelihood methods are easy to apply to misclassification 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 mismeasured independent variables. 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 the MI approach) 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. However, specifying prior distributions introduces additional methodological complexity and posterior inference is computationally intensive, making Bayesian methods less convenient for Monte-Carlo simulation.

Appendix D

Deriving the Maximum Likelihood Approach

In the following, we derive our MLE approach for addressing misclassifications.

When an AC Measures an Independent Variable

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.

$$P(Y, W) = \sum_x P(Y, W, X = x) \tag{D1}$$

$$= \sum_x P(Y|W, X = x)P(W, X = x) \tag{D2}$$

$$= \sum_x P(Y, X = x)P(W|Y, X = x) \tag{D3}$$

$$= \sum_x P(Y|X = x)P(W|Y, X = x)P(X = x) \tag{D4}$$

Equation D1 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 D2 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 automated 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 D3 shows a different way to factor the joint probability $P(Y, W)$ so that W is not in the model of Y. Since X and W are correlated, if W is in the model for Y, the

estimation of B_1 will be biased. By including Y in the model for W, Equation D3 can account for differential measurement error.

Equation D4 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 D1–D4 demonstrate the generality of this method because the conditional probabilities may be calculated using a wide range of probability models. For simplicity, we have focused 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 Dependent Variable

Again, we will maximize $\mathcal{L}(\Theta|Y, W)$, the joint likelihood of the parameters Θ given the outcome Y and automated classifications W measuring the dependent variable Y (Carroll et al., 2006). 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 $P(W|Y)$, the conditional probability of W given Y.

$$P(Y, W) = \sum_y P(Y = y, W) \tag{D5}$$

$$= \sum_y P(Y)P(W|Y) \tag{D6}$$

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. It can fit linear and logistic regression models when an AC measures an independent variable and logistic regression models when an AC measures the dependent variable. 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).

Comment on model assumptions

How burdensome is the assumption that the error model be able to consistently estimate the conditional probability of W given Y? If this assumption were more difficult than those already accepted by the model for Y given X and Z, one would fear that using the MLE correction method introduces greater validity threats than it removes. However, the assumption that a model for Y given X is consistent implies that the necessary variables to model the dependence of W on Y and X are observed.

This is clear because if our model for $P(Y|X,Z)$ has no omitted variables (i.e., $P(Y|X,Z) = P(Y|X,Z,T)$ for any unobserved variable T correlated with X or Z) then we can be sure that our model for $P(Y,W)$ does not either. To see why, suppose T is an unobserved variable that is not an omitted variable from $P(Y|X,Z)$. By conditional probability, $P(W|T,X,Y,Z) = \frac{P(T,W,X,Y,Z)}{P(T,X,Y,Z)} = \frac{P(Y|T,W,X,Z)P(T,W,X,Z)}{P(Y|T,X,Z)P(T,X,Z)}$. Either T is uncorrelated with X, Z, or Y, in which case it is not an omitted variable from $P(W|X,Y,Z)$ or $P(Y|X,Z) = P(Y|T,X,Z)$. Assuming the later, $\frac{P(Y|T,W,X,Z)P(T,X,W,Z)}{P(Y|T,X,Z)P(T,X,Z)} = \frac{P(T,W,X,Z)}{P(T,X,Z)} = P(W|X,Y,Z)$ (note that W is not an omitted variable from $P(Y|X,Z)$). As a result, $P(W|T,X,Y,Z) = P(W|X,Y,Z)$ and T is not omitted from

our model for $P(W|X, Y, Z)$.

In sum, if one can assume a model for $P(Y|X, Z)$, one can assume the variables needed to model $P(W|X, Y, Z)$ are observed. It is important to include all such variables in the error model as demonstrated in Appendix F. In general, including all variables in the model for Y in the model for W is a good default. Specifying the functional form of the relationship may will require care around nonlinearities, but is feasible.

Appendix E

misclassificationmodels: The R package

The package provides a function to conduct regression analysis but also corrects for misclassification using information from manually annotated 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 that w is the proxy of the ground truth x.
- The manually annotated data must be provided via the argument *data2*

The following snippet shows a typical scenario, here for correcting misclassifications in an independent variable:

```
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

For more information about the package, please see here: https://osf.io/pyqf8/?view_only=c80e7b76d94645bd9543f04c2a95a87e.

Appendix F

Additional Plots and Simulations

In addition to the results reported in the main paper, we include in the next section auxiliary plots from the main simulations. Below, we present results from further simulations that show what happens when the error model is mispeccified, how results vary with classifier predictivness or when the classified variable is not balanced, but skewed, and as the degree to which misclassification is systematic varies. a

Additional plots for Simulations 1 and 2

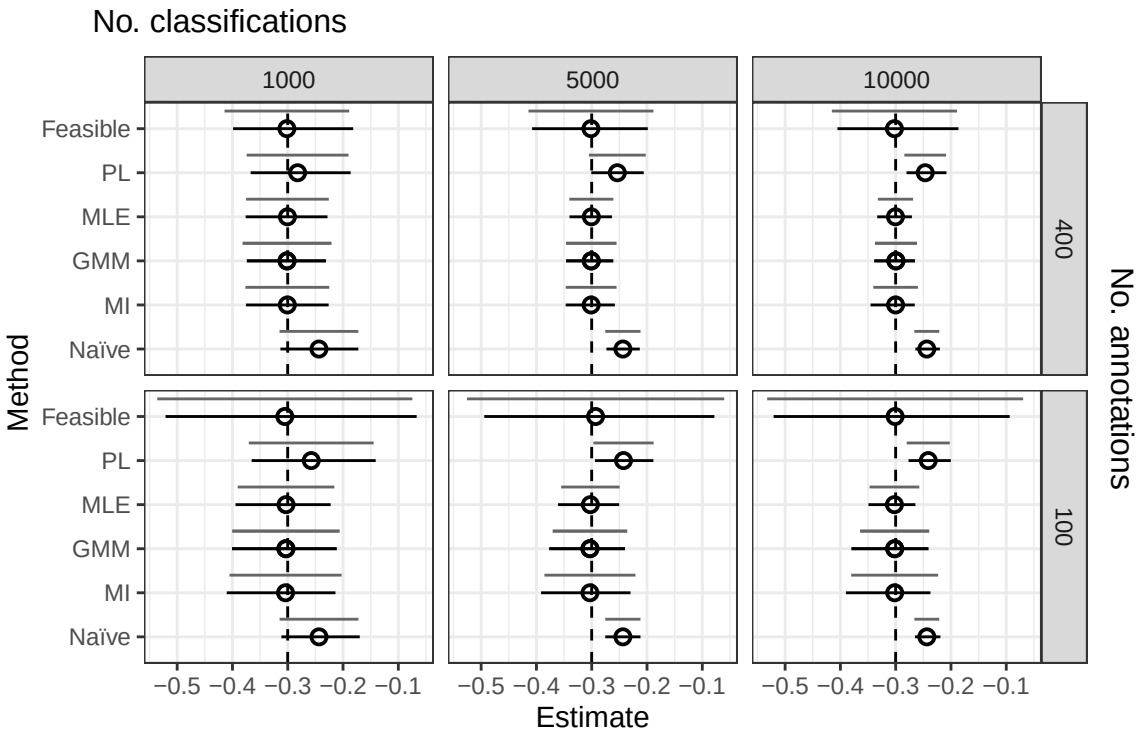


Figure F1

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.*

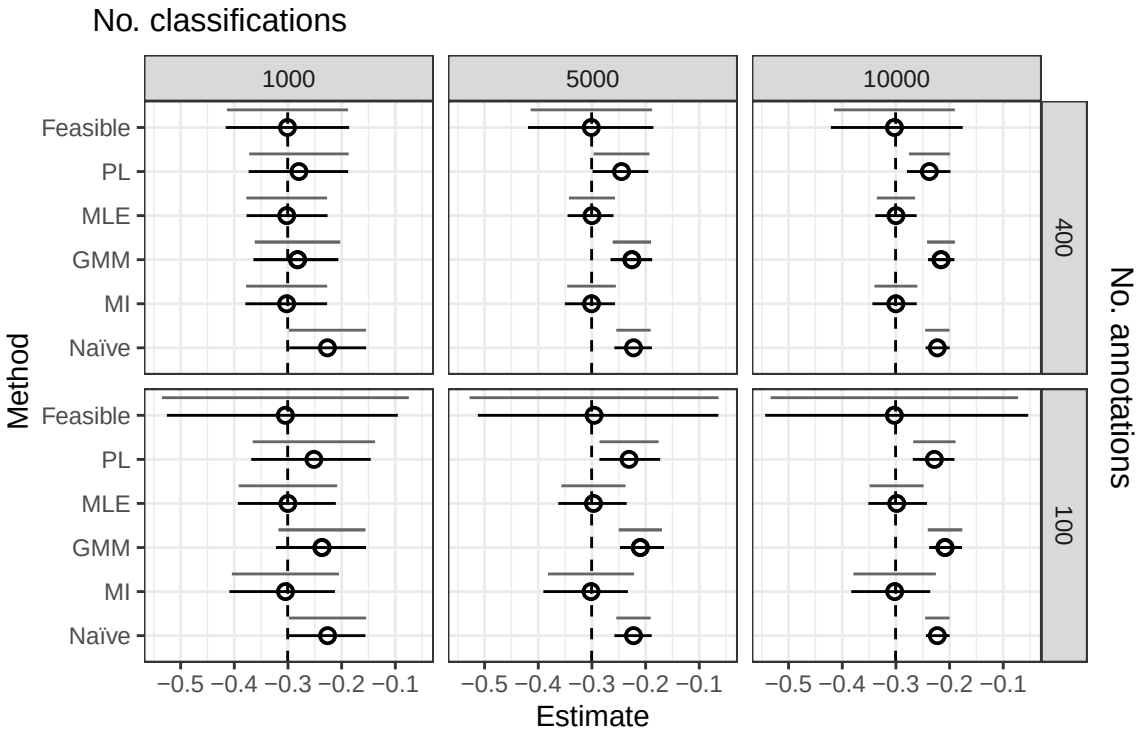


Figure F2

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.

Simulating what happens when an error model is misspecified.

In simulations 1b and 2b, the MLE method was able to correct systematic misclassification using the error models in equations 1 and 5. However, this depends on the error model consistently estimating the conditional probability of automatic classifications given the true value and the outcome. If the misclassifications and the outcome are conditionally dependent given a variable Z that is omitted from the model, this will not be possible. Here, we demonstrate how such misspecification of the error model can affect results.

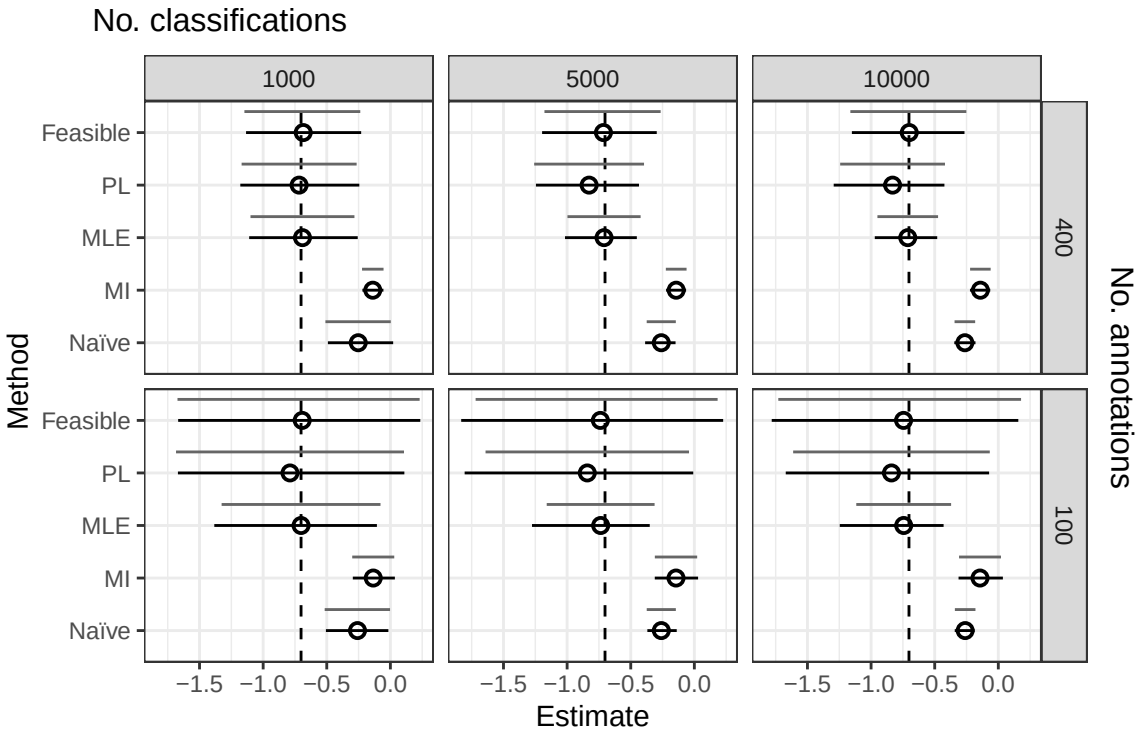


Figure F3

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.*

Systematic Misclassification of an Independent Variable with Z omitted from the error model

What happens in simulation 1b, representing systematic misclassification of an independent variable, when the error model is missing variable Z? As shown in Figure F5 this incorrect MLE model is unable to fully correct misclassification bias. Although the estimate of B_X is close to correct, estimation of B_Z is clearly biased, if improved compared to the näive estimator.

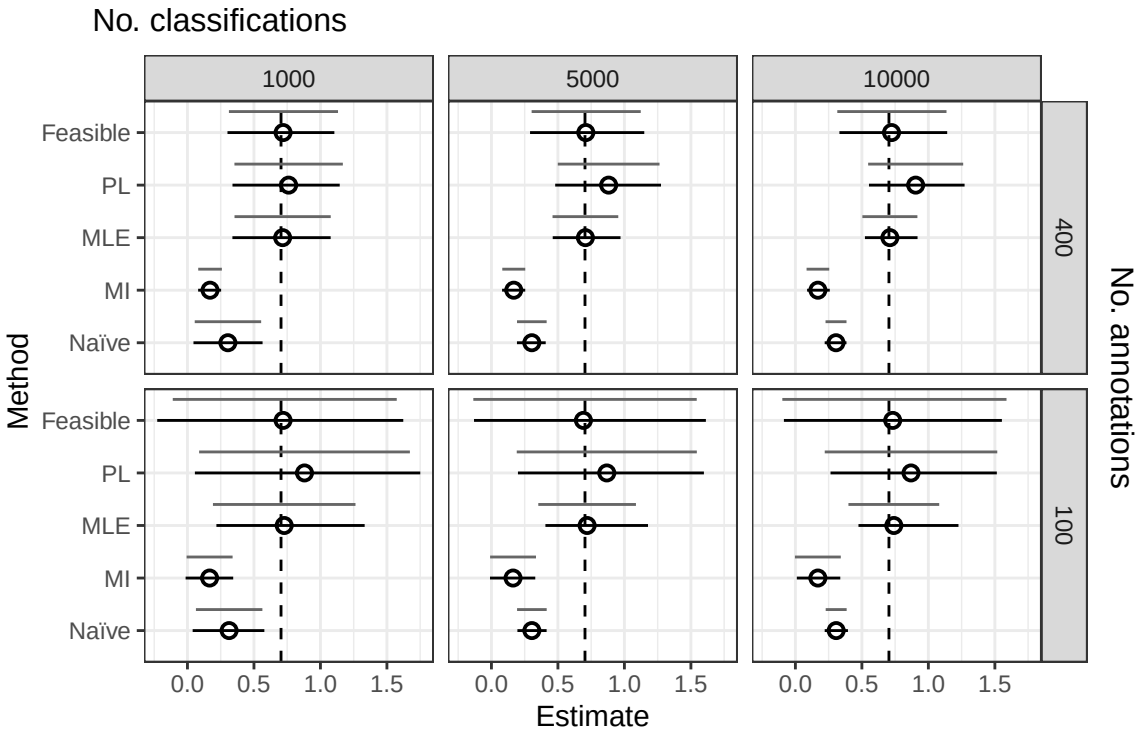


Figure F4

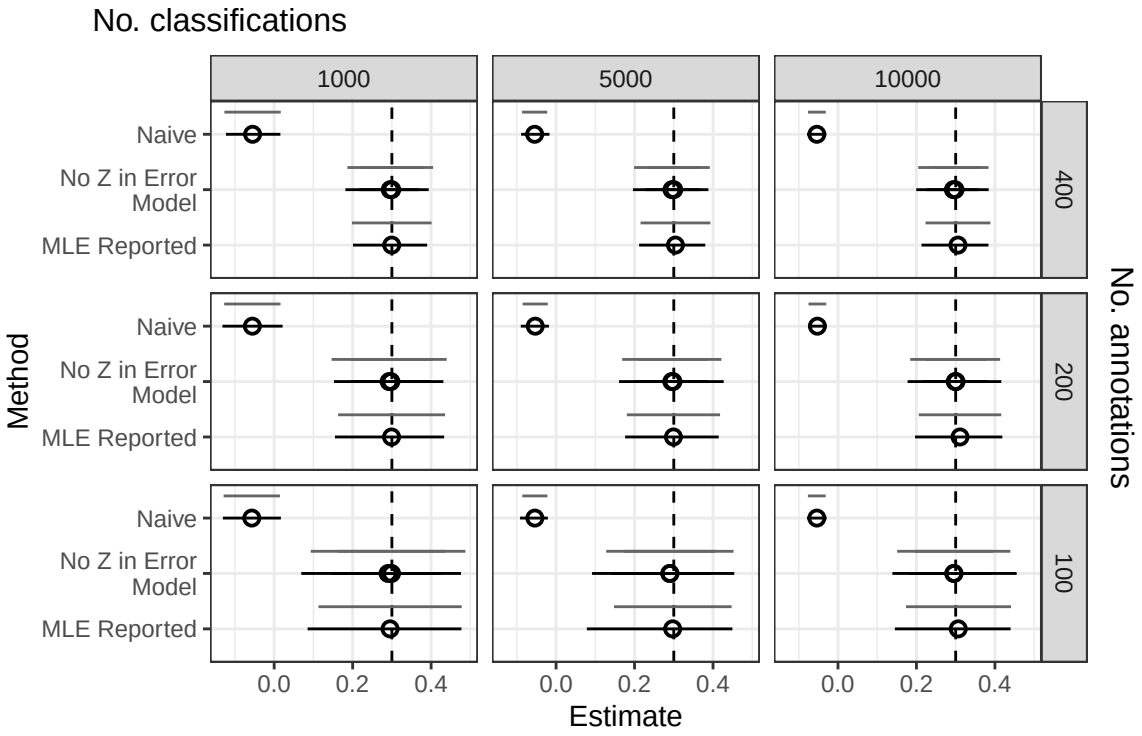
Estimates of B_X in simulation 2b *multivariate regression with Y measured using machine learning, model accuracy correlated with Z and Y and differential error. Only our MLE model with a full specification of the error model obtains consistent estimates.*

Systematic misclassification of a dependent variable with Z omitted from the error model.

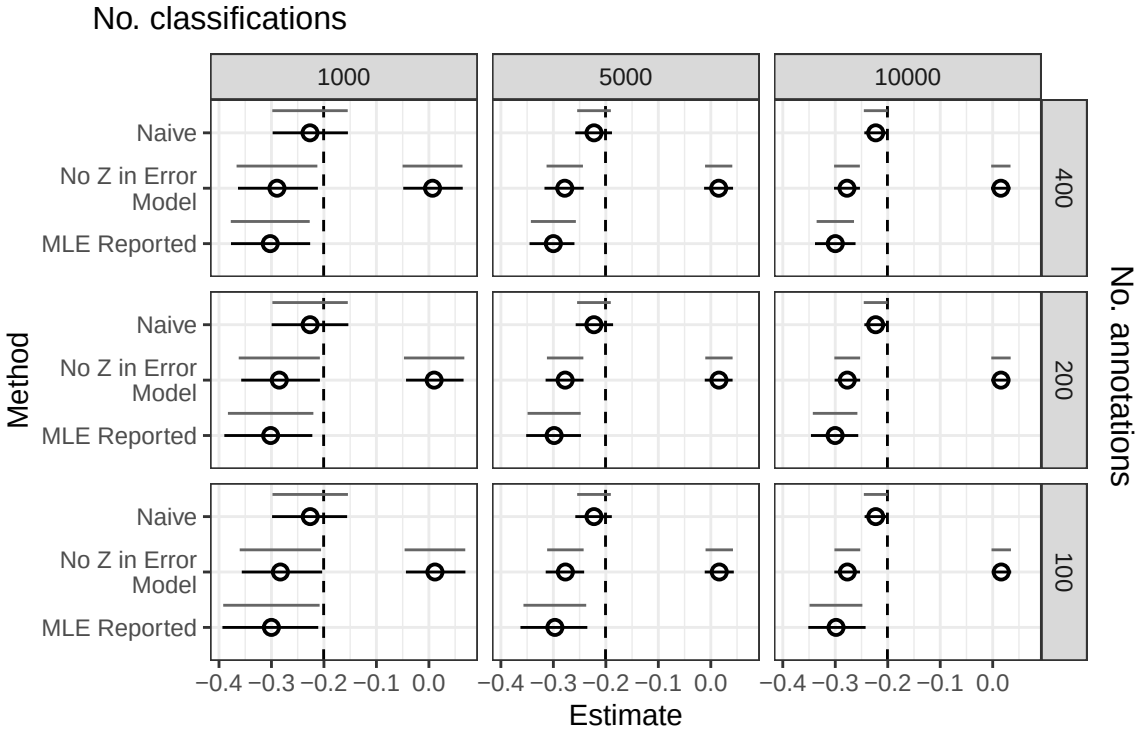
Similarly, as shown in Figure F6, in the case a dependent variable is systematically misclassified, an error model omitting a variable Z required to make W and Y conditionally independent is unable to obtain consistent estimates. Again, the estimate of B_X is close to the true value, but the estimate of B_Z is biased, if less so than the näive estimate.

Simulating varying automatic classifier accuracy

To explore how misclassification bias and correction methods depend on classifier performance, we repeat Simulations 1a and 1b with levels classifier accuracy ranging from 60% to 95%. We present results with the sample size with 5,000 classifications and 100



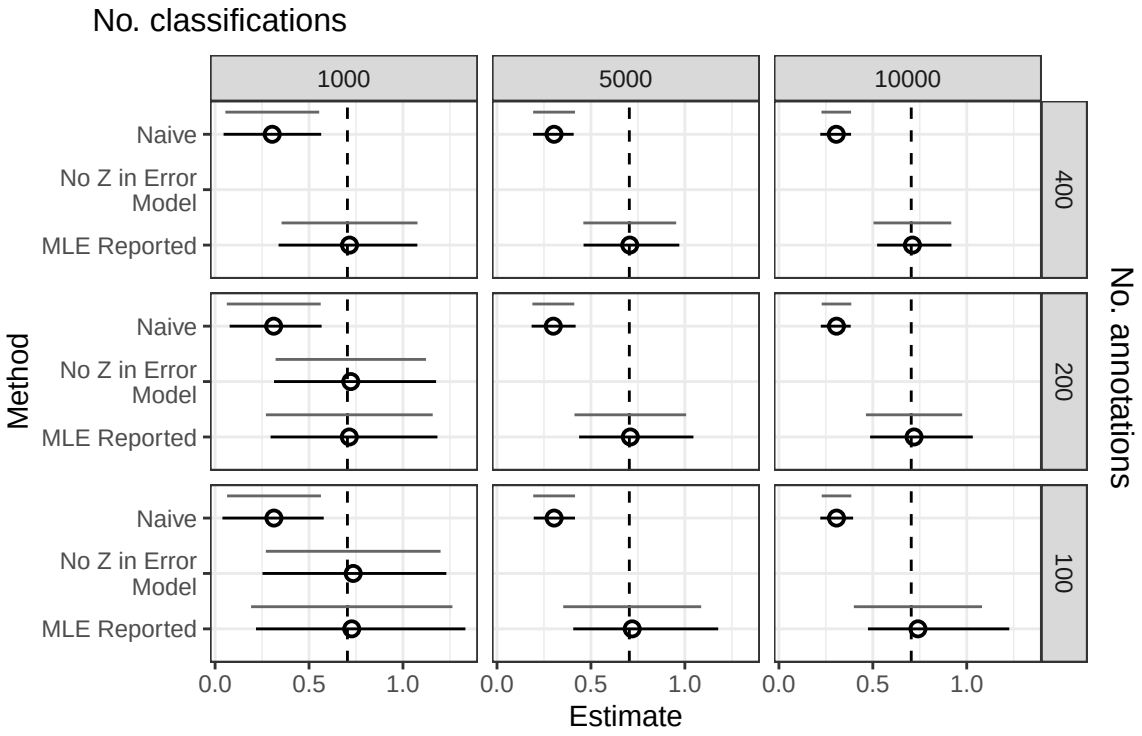
(a) *Estimates of B_X with a misspecified error correction model that omits Z are still close to the true value.*



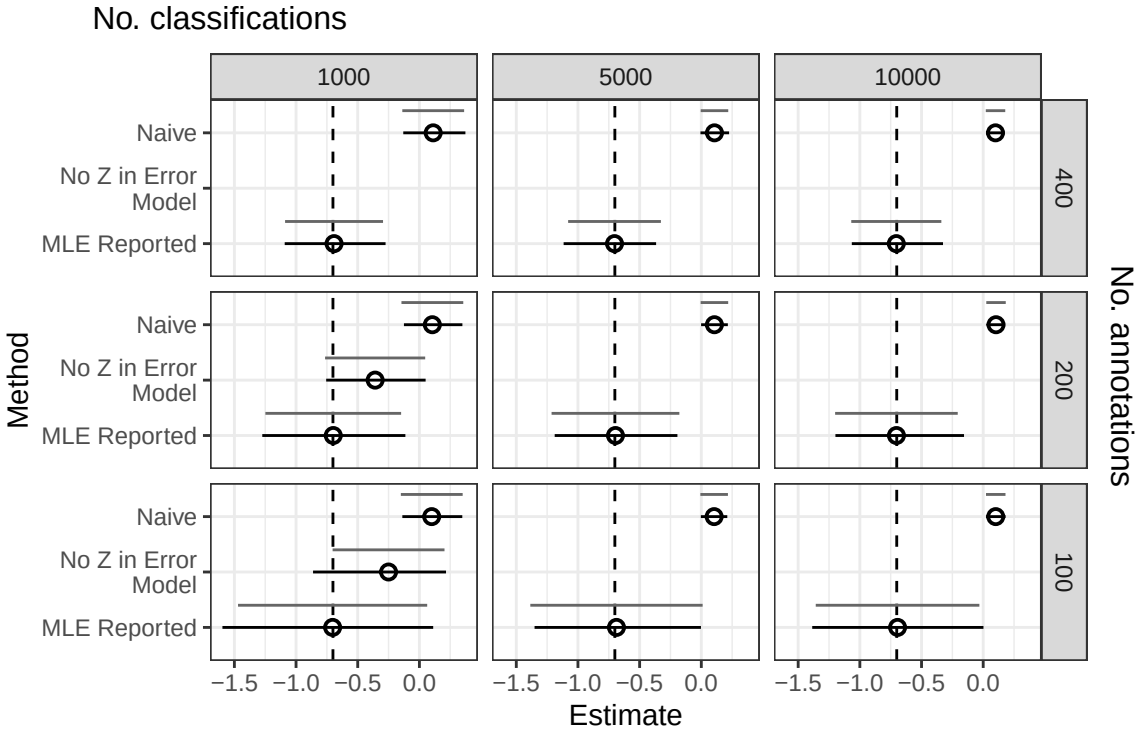
(b) *Estimates of B_Z with a misspecified error correction model that omits Z are noticeably biased but better than the näive estimator.*

Figure F5

Failure to correct for misclassification in an independent variable when the error model is misspecified.



(a) *Estimates of B_X with a misspecified error correction model that omits Z are still close to the true value.*



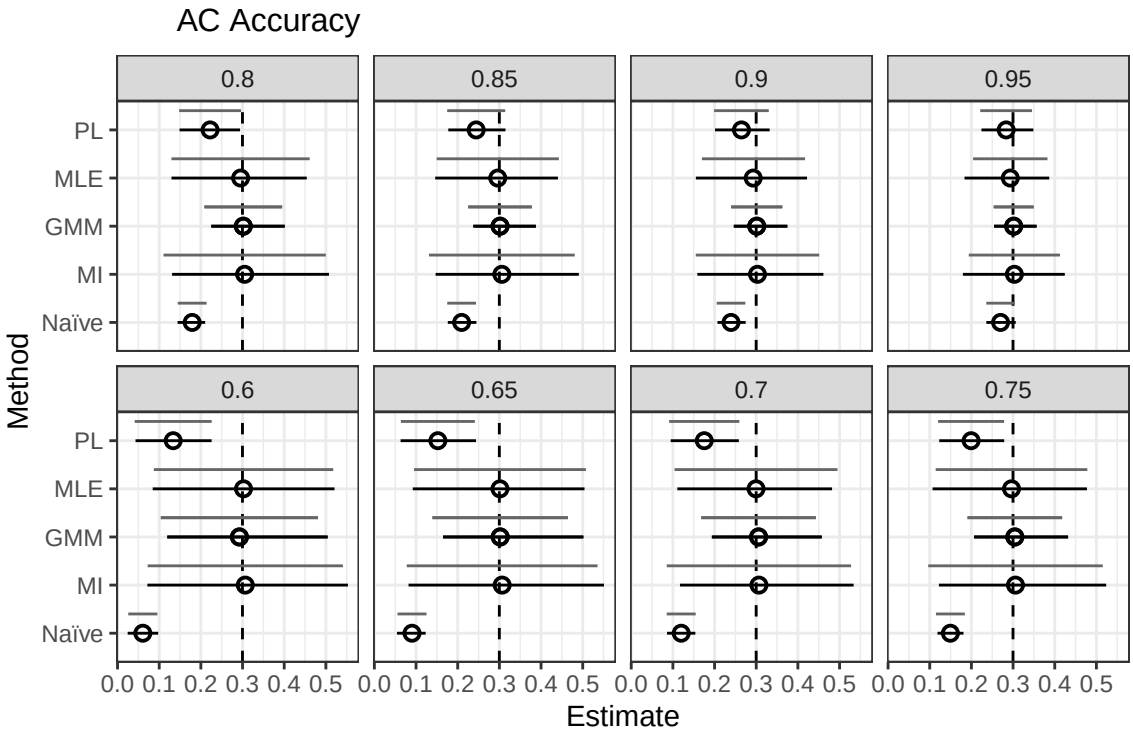
(b) *Estimates of B_Z with a misspecified error correction model that omits Z are noticeably biased but better than the näive estimator.*

Figure F6

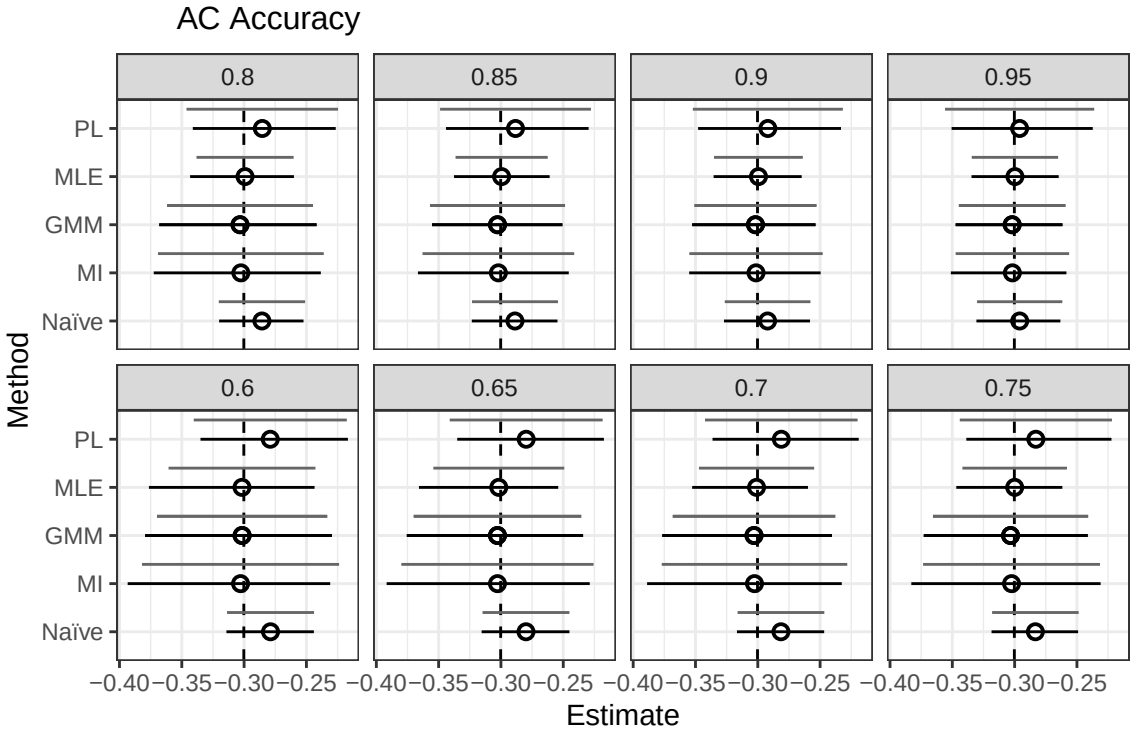
Failure to correct for misclassification in an independent variable when the error model is misspecified

annotations. As expected, in both scenarios a more accurate classifier causes less misclassification bias. All the error correction methods provide more precise estimates when used with a more accurate classifiers.

Simulating misclassification in skewed variables



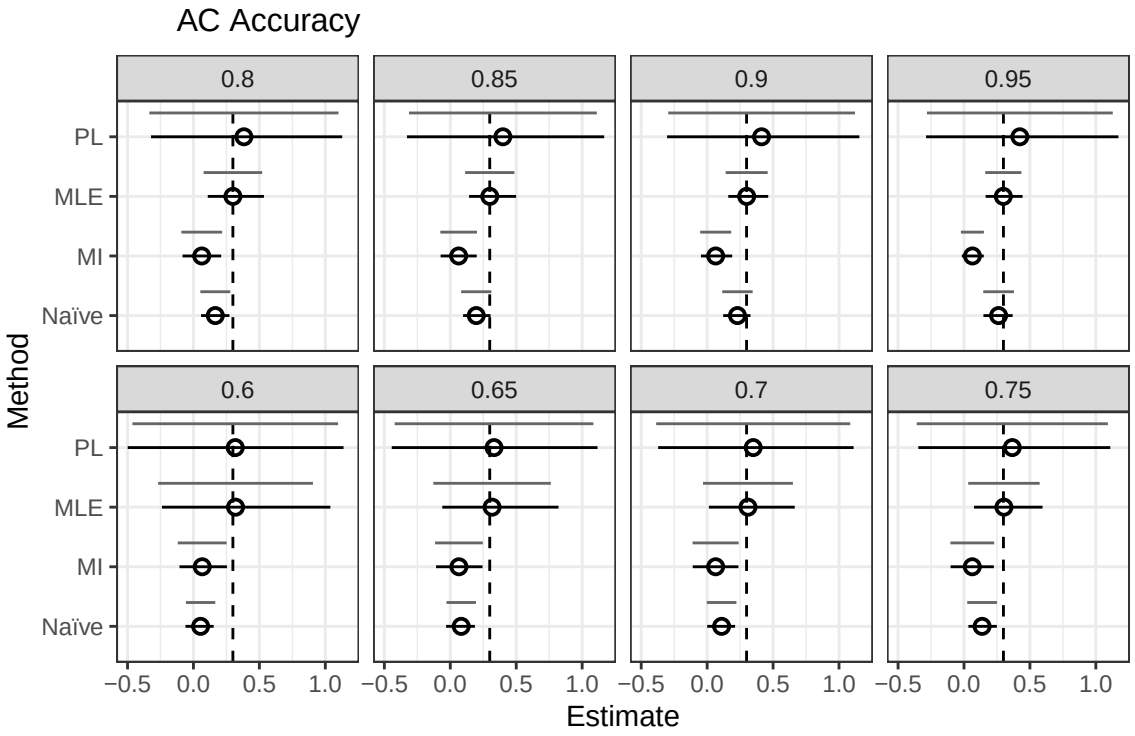
(a) *Estimates of B_X with a misspecified error correction model that omits Z are still close to the true value.*



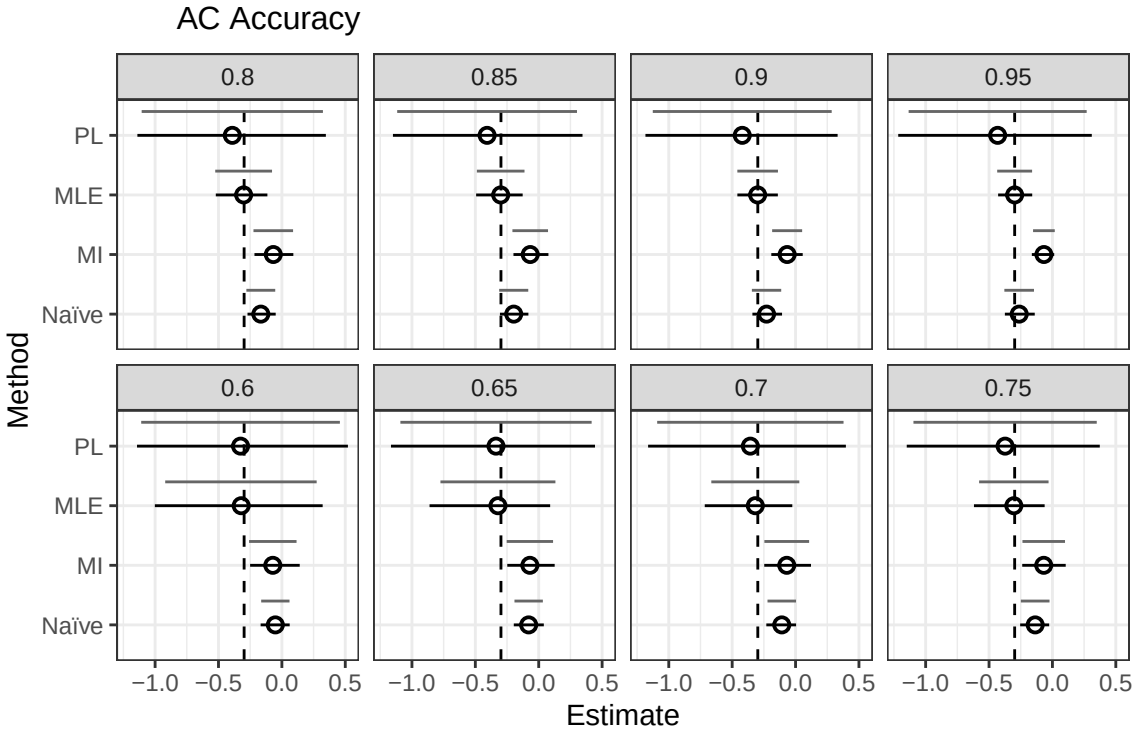
(b) *Estimates of B_Z with a misspecified error correction model that omits Z are noticeably biased but better than the näive estimator.*

Figure F7

Failure to correct for misclassification in an independent variable when the error model is misspecified



(a) *Estimates of B_X with a misspecified error correction model that omits Z are still close to the true value.*



(b) *Estimates of B_Z with a misspecified error correction model that omits Z are noticeably biased but better than the näive estimator.*

Figure F8

Failure to correct for misclassification in an independent variable when the error model is misspecified.