Online Appendix For:

**In Validations We Trust? The Impact of Imperfect Human Annotations as a Gold Standard on the Quality of Validation of Automated Content Analysis**

**1. Variables coded in Study 1, detailed coding instructions, and reliability estimates**

Using EBSCOhost databases, we searched all English-language journal articles published between January 1, 1998 and November 7, 2018, querying all titles, abstracts, and keywords using the following Boolean search string: ("computer assisted" OR "automated" OR "automatic" OR "computational" OR "machine learning") AND ("content analysis" OR "text analysis") This was done by examining “Communication & Mass Media Complete,” “Humanities Source,” and “SocINDEX with Full Index” collections.

Among a total of 192 retrieved articles, 112 articles were determined as not relevant (e.g., non-empirical overviews/introduction articles, qualitative analyses, studies using unsupervised methods, or simple keyword frequencies, etc.) and 7 articles were either duplicates or could not be obtained as full texts. These articles were excluded from further analyses. Here, we exclude a simple keyword-frequency based study (e.g., simply counting the number of occurrences of a keyword in a given text, but not actually classifying the documents based on such frequency) since human inputs play no role other than compiling the keyword list itself. Among excluded studies, only 15 studies have used unsupervised learning or other forms of automated content analysis, suggesting dictionary-based or supervised machine learning applications are much more frequently used in general.

A total of five highly trained coders tested the initial coding scheme by independently coding 10 randomly sampled articles (approximately 5% of the total retrieved sample, N = 119) and collectively discussed any coding problems and disagreements. Traditional content analysis literature generally recommends 5% to 25% of all materials to be used for reliability assessment (Lacy & Riffe, 1996). Coding instructions were iteratively revised until the coding schemes would produce reliable results. Intercoder reliability (based on Krippendorff’s alpha) above 0.75 was ensured for each of the variables coded. Following variables were independently coded by 5 trained coders.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Definition & Coding instructions** | **Reliability** |
| Relevance | Whether empirical text analysis is conducted and reported (Yes = 1, No = 0) | Alpha = 1 |
| Method Used | 1 = Search string based / Dictionary Approach  2 = Machine Learning  3 = Topic Modeling (excluded from further analysis)  4 = Other (excluded from further analysis) | Alpha = 1 |
| Refer to gold standard | 1 = Yes, a “gold standard” is used, and info is reported  0 = No is not used reported | Alpha = 1 |
| Report reliability | Whether intercoder-reliability of human-coded materials are reported?  (1 = Yes, reported, 0 = Not reported) | Alpha = 1 |
| Refer to validation / Report validation measures | Whether validation of automated procedures are mentioned, and if so, whether either one of validation metrics (e.g., Recall, Sensitivity, Precision, Accuracy, F1, or other measures) is reported?  (1 = Yes, mentioned, 0 = Not mentioned) | Alpha = .750 |

**2. Detailed Setup of MC simulations**

**Data Generation**

We create data (e.g., textual data, such as newspaper articles, to be analyzed) with the “true” outcome value of interest, y (i.e., a classification membership of a given document); the goal of any quantitative text analysis method is to somehow directly approximate this value of y for each observation-level, or instead estimate the unbiased distribution of y at the aggregate level (Grimmer & Stewart, 2013). For the data generating process, we set y at each document level to be randomly generated from three hypothetical independent variables (x1, x2, and x3), all of which stand for some textual features (e.g., words or phrases) of a given document, plus a certain unobserved feature (x0) that is not evenly distributed across the dataset. The values of those variables were randomly sampled from a multivariate normal distribution. In addition, values of x0 were set to be identical across certain grouping variables of media content data, effectively simulating features that are not randomly nor uniformly distributed in the data. This ensures that the results of our simulations are not completely deterministic nor analytically driven to arrive at our conclusion.

**SML Scenario.** For supervised machine learning approach, we set the true values of (which is the binary variable) are sampled from a Binomial distribution, with the probability parameter having a very simple linear functional form as follows:

with being Gaussian noise added to ensure that each simulation run is not completely deterministic. The , the true population parameter, was fixed throughout the simulation runs (specifically, = 1, = 0.5, = 0.2, and = 0.6, which were randomly chosen). Following this setup, a single simulation run is set to generate a total of 130,000 observations of media content data.

**Dictionary-based Scenario.** For a dictionary (i.e., bag-of-words) method, we assume a very similar approach as discussed above, but additionally truncate the values of independent variables to its nearest integer values (i.e., a discrete value), where they represent some “features” of given textual data (e.g., a word) or a combination of such textual features (e.g., a word order or N-grams), in a similar fashion as in Equations (1). Yet for the dictionary-based approach, the vector was extended to *K* = 5 and their values were fixed to 0.2. This enables us to better approximate the multidimensionality of textual data, while treating effectively as a function of the simple sum of the chosen textual features (which is a general assumption that most of the dictionary-based classification methods assumes).

This slight modification for dictionary approaches – truncating to the nearest integers – is due to the fact that each “feature” in the text (e.g., words, phrases, or boolean expressions, etc.) should be “predefined” to be matched against identical forms of dictionaries. We therefore effectively treat simulated integer numbers for three independent variables as each of the predefined categories for textual features, whose scores are simply taken from the existing dictionaries based on some rules. In contrast, for SML scenarios, we use raw continuous normal distributions as is (without rounding up/down numbers) effectively treating them as some kind of a transformed vector dimensional space wherein algorithms try to separate the observations into two categories (i.e., classification membership to be estimated) on that space.

**Human Coding**

In all scenarios, human coders classify a given observation as “1” (e.g., a text contains the quantity of interest, such as a certain actor, frame, or tonality) or “0” (e.g., does not contain this quantity), based on some observable features of each documents. This human coding (y) can be, in principle, either correct or incorrect against the (unknown) true value, y, therefore behaviors of human coders were modeled by a Binomial distribution with varying probability of successfully categorizing the true data. This enables us to simulate a situation where, at a given target reliability level, some coders produce “correct” judgments while other coders produce “false” judgments more often.

**Algorithm-based Classification and Validation**

For the dictionary approach, we assume that a researcher utilizes an off-the-shelf dictionary, based on mean valence of observed textual features (e.g., words, phrases, etc.), whose valence scores are taken from the existing dictionary. For the SML approach, we also assume that appropriate, domain-specific annotated materials for a given task already exist for the algorithm development, with a fixed number of training materials (*N* = 5000, approximately 4% of the total dataset being coded).[[1]](#footnote-1)

**3. Additional results referred in the main results.**

![A close up of a pencil

Description automatically generated]()

Figure A1. Mean Absolute Prediction Error (point estimate) and their 68% (±1SD) and 95% (±2SD) percentile intervals for every combination of experimental factors, **SML** scenarios (*N* = 1,000 per scenario).

![A screenshot of a cell phone

Description automatically generated]()

Figure A2. Mean Absolute Prediction Error (point estimate) and their 68% (±1SD) and 95% (±2SD) percentile intervals for every combination of experimental factors, **dictionary** scenarios (*N* = 1,000 per scenario).

**ANOVAs estimating interactions of *number of coders*, *duplicated codings*, and *intercoder reliability* with other factors, SML scenarios.**

A 3-way interaction among *intercoder reliability, size of dataset, and random sample*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factors** | **Df** | **SS** | **MS** | **F** | **Pr(>*F*)** |
| No. of coders | 2 | .00 | .00 | .096 | .908 |
| Duplicated vs. Sole-coding | 1 | .00 | .00 | .012 | .913 |
| Size of validation data (N) | 3 | .0013 | .0004 | 359.497 | .001 \*\*\* |
| Target Krippendorff’s alpha (K) | 2 | .0207 | .0104 | 8036.286 | .001 \*\*\* |
| Random sample vs. not (R) | 1 | .0084 | .0084 | 6532.416 | .001 \*\*\* |
| K \* N | 6 | .0004 | .00006 | 52.884 | .001 \*\*\* |
| N \* R | 3 | .0001 | .00004 | 35.610 | .001 \*\*\* |
| K \* R | 2 | .0021 | .0010 | 807.706 | .001 \*\*\* |
| K \* N \* R | 6 | .00004 | .000007 | 5.551 | .001 \*\*\* |
| Residuals | 117 | .00015 | .000001 |  |  |

A 2-way interaction with the *number of coders*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factors** | **Df** | **SS** | **MS** | **F** | **Pr(>*F*)** |
| No. of coders (k) | 2 | .00 | .00 | .005 | .995 |
| Duplicated vs. Sole-coding | 1 | .00 | .00 | .001 | .979 |
| Size of validation data | 3 | .0013 | .0004 | 19.853 | .001 \*\*\* |
| Target Krippendorff’s alpha | 2 | .0207 | .0103 | 443.801 | .001 \*\*\* |
| Random sample vs. not | 1 | .0084 | .0084 | 360.750 | .001 \*\*\* |
| k \* Duplicated vs. Sole-coding | 2 | .000005 | .000002 | .106 | .900 |
| k \* Size of validation data | 6 | .000011 | .000002 | .076 | .998 |
| k \* Krippendorff’s alpha | 4 | .000004 | .000001 | .041 | .977 |
| k \* Random sample vs. not | 2 | .000002 | .000001 | .053 | .949 |
| Residuals | 120 | .002803 | .000023 |  |  |

A 2-way interaction with *duplicated coding*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factors** | **Df** | **SS** | **MS** | **F** | **Pr(>*F*)** |
| No. of coders | 2 | .00 | .00 | .006 | .994 |
| Duplicated vs. Sole-coding (D) | 1 | .00 | .00 | .001 | .979 |
| Size of validation data | 3 | .0013 | .0004 | 20.769 | .001 \*\*\* |
| Target Krippendorff’s alpha | 2 | .0207 | .0103 | 464.284 | .001 \*\*\* |
| Random sample vs. not | 1 | .0084 | .0084 | 377.400 | .001 \*\*\* |
| D \* No. of coders | 2 | .000005 | .000002 | .111 | .895 |
| D \* Size of validation data | 3 | .000005 | .000002 | .079 | .971 |
| D \* Krippendorff’s alpha | 2 | .000001 | .00 | .022 | .978 |
| D \* Random sample vs. not | 1 | .00 | .00 | .011 | .918 |
| Residuals | 126 | .002813 | .000022 |  |  |

1. This means that researchers would only require to produce human coding for validation materials. In practice, when domain-appropriate training materials are not available, one need to produce human coding for training/testing materials as well. Doing so means the “quality” of human coding in such training/testing materials would be the same as validation materials, since one rarely employ different standards for training/testing vs. validation materials in such cases [↑](#footnote-ref-1)