Psychological Methods

The Repeated Adjustment of Measurement Protocols Method for Developing High-Validity Text Classifiers

Alex Goddard and Alex Gillespie

Online First Publication, October 6, 2025. https://dx.doi.org/10.1037/met0000787

CITATION

Goddard, A., & Gillespie, A. (2025). The repeated adjustment of measurement protocols method for developing high-validity text classifiers. *Psychological Methods*. Advance online publication. https://dx.doi.org/10.1037/met0000787

AMERICAN PSYCHOLOGICAL ASSOCIATION

Psychological Methods

ISSN: 1082-989X

https://doi.org/10.1037/met0000787

The Repeated Adjustment of Measurement Protocols Method for Developing High-Validity Text Classifiers

Alex Goddard[1] and Alex Gillespie[1, 2]

[1] Department of Psychological and Behavioural Science, London School of Economics and Political Science

[2] Department of Psychology, Oslo New University College

Abstract

The development and evaluation of text classifiers in psychology depends on rigorous manual coding. Yet, the evaluation of manual coding and computational algorithms is usually considered separately. This is problematic because developing high-validity classifiers is a repeated process of identifying, explaining, and addressing conceptual and measurement issues during both the manual coding and classifier development stages. To address this problem, we introduce the Repeated Adjustment of Measurement Protocols (RAMP) method for developing high-validity text classifiers in psychology. The RAMP method has three stages: manual coding, classifier development, and integrative evaluation. These stages integrate the best practices of content analysis (manual coding), data science (classifier development), and psychology (integrative evaluation). Central to this integration is the concept of an inference loop, defined as the process of maximizing validity through repeated adjustments to concepts and constructs, guided by push-back from the empirical data. Inference loops operate both within each stage of the method and across related studies. We illustrate RAMP through a case study, where we manually coded 21,815 sentences for misunderstanding (Krippendorff's $\alpha = .79$), and developed a rule-based classifier (Matthews correlation coefficient [MCC] = 0.22), a supervised machine learning classifier (Bidirectional Encoder Representations From Transformers; MCC = 0.69) and a large language model classifier (GPT-4o; MCC = 0.47). By integrating manual coding and classifier development stages, we were able to identify and address a concept validity problem with misunderstandings. RAMP advances existing methods by operationalizing validity as an ongoing dynamic process, where concepts and constructs are repeatedly adjusted toward increasingly widespread intersubjective agreement on their utility.

Translational Abstract

Text classifiers are algorithms that sort documents into categories. Modern text classifiers leverage artificial intelligence (AI) and are tested by comparing their categorizations with those produced by human coders. However, this human coding stage is rarely considered systematically. This is problematic because poor-quality human classifications can be accurately reproduced by algorithms (garbage in, garbage out), creating the illusion of a valid measure. To address this problem, we introduce the Repeated Adjustment of Measurement Protocols (RAMP) method for developing and testing text classifiers. The method is designed to ensure meaningful and accurate measurements of psychological concepts (e.g., personality and communication behaviors) in text. It formally integrates the best practices of content analysis (manual coding), data science (classifier development), and psychology (integrative evaluation). We illustrate RAMP through a case study on measuring misunderstandings in online dialogues. We developed three different text classifiers using RAMP: one that counts words to identify misunderstandings, one that learns to reproduce manual coding using human examples, and one that leverages an AI chatbot (OpenAI's ChatGPT). By analyzing the problems in human coding and text classifier output, our case study revealed problems with the underlying concept of misunderstanding. RAMP reveals that high-validity measurement is an ongoing process, where guiding concepts and constructs are continually being updated. To support the use of RAMP, we provide a checklist for developing text classifiers, alongside all code and data for replicating the results reported in the case study.

Douglas Steinley served as action editor.

Alex Goddard https://orcid.org/0000-0003-1382-2700

Alex Gillespie https://orcid.org/0000-0002-0162-1269

The authors have no known conflict of interests to disclose. All data have been made publicly available as the additional online materials on the Open Science Framework (Goddard & Gillespie, 2025a) and can be accessed at https://osf.io/pe4jy/. The code behind the analysis has been made publicly available on GitHub (Goddard & Gillespie, 2025b) and can be accessed at https://github.com/alexiamhe93/RAMP_method.

Open Access funding provided by London School of Economics and Political Science: This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0; https://creativecommons.org/licenses/by/4.0). This license permits copying and redistributing the work in any medium or format, as well as adapting the material for any purpose, even commercially.

Alex Goddard served as lead for conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, validation, visualization, writing–original draft, and writing–review and editing. Alex Gillespie served as lead for resources and supervision and served in a supporting role for conceptualization, methodology, and writing–review and editing.

Correspondence concerning this article should be addressed to Alex Goddard, Department of Psychological and Behavioural Science, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, United Kingdom. Email: a.j.goddard@lse.ac.uk

Keywords: text classification, construct validity, content analysis, large language models, conceptual clarity

Supplemental materials: https://doi.org/10.1037/met0000787.supp

Advanced text classifiers are increasingly used in psychological research. However, there are few conventions for establishing their validity (Birkenmaier et al., 2024). Rigorous manual coding is an integral stage in developing a valid text classifier, yet this process is rarely scrutinized (Song et al., 2020). This is problematic because text classifiers can reliably and accurately reproduce low-validity manual coding. To address this problem, we integrate the best practices for manual coding, classifier development, and psychometric validation into a single iterative framework: the Repeated Adjustment of Measurement Protocols (RAMP) method.

The RAMP method has three stages. The first stage employs conventions from content analysis to generate a manually coded data set (Krippendorff, 2019). The second stage uses a data science approach to developing text classifiers, relying on withholding test data during development (Donoho, 2017). The third stage assesses the concept and construct validity of the classifier using an abductive process, grounded in psychometric validation (Gillespie et al., 2024). While each stage has been discussed in the literature, they have been conceptualized separately. RAMP integrates these stages, revealing the importance of feedback from the latter stages (i.e., classifier development and integrative evaluation) to the initial stage (i.e., conceptualization, manual coding). This backward propagating feedback, we argue, is central to developing high-validity classifiers.

The engine of RAMP is the inference loop in which induction, deduction, and abduction combine to incrementally increase the construct validity of the classifier, and the validity of the underlying concept. The inference loop underlies all three stages of RAMP, can be applied to developing different types of classifiers, and conceptualizes the process by which measurements and concepts are refined over time across multiple studies.

We illustrate RAMP using a case study on measuring misunderstandings (Laing et al., 1966) in social media data. We employed the inference loop for manual coding, developing three different text classifiers (rule-based, supervised, and large language model [LLM]), and for an integrative evaluation, where we uncovered a concept validity problem with misunderstandings. To evaluate the effectiveness of iterative inference loops, we compared our results to a noniterative approach using the same data. We found that the RAMP method generates more reliable manual coding and more accurate text classifiers. The case study thus showcases how inference loops, which reconcile theory with data, power each stage of the method. We provide all code and data for replicating the study, and a checklist for employing and reporting the RAMP method (Appendix).

Validity in Psychological Text Classification

Validity is central to all measurement. RAMP focuses on two key forms of validity: concept and construct validity. A concept is defined as any term or idea that relates to a theory or model (Bringmann et al., 2022). Concept validity is the degree to which a concept is justified by a robust theoretical framework (Locke, 2012). A construct is defined as any operationalization of a concept through a measurement protocol. Construct validity refers to how well a protocol represents concepts through its constructs (Cronbach & Meehl, 1955).

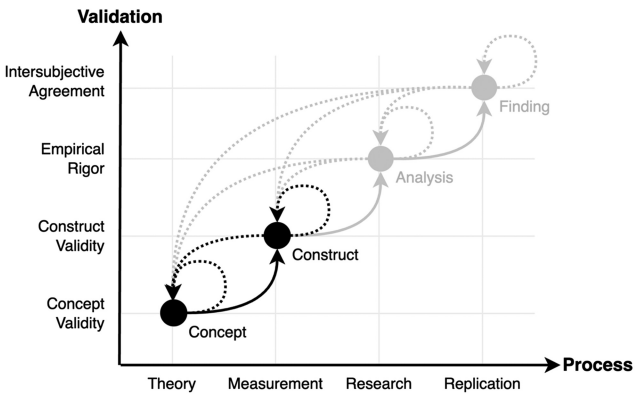
Concept and construct validity are prerequisites to conducting rigorous empirical research and generating replicable findings (Flake & Fried, 2020; Flake et al., 2022). The goal of science is to establish intersubjective agreement about empirical findings, defined as the process of scaffolding objectivity through the collective consensus of researchers (Freeman, 1973; Peirce, 1955; Popper, 1959). In psychology, intersubjective agreement on findings was undermined when results, previously thought valid, failed to replicate (Open Science Collaboration, 2015). These failures were initially attributed to poor statistical rigor (e.g., Nosek et al., 2018); however, recent literature has also pointed to concept and construct validity issues (Bringmann et al., 2022; Eronen & Bringmann, 2021; Flake & Fried, 2020).

We introduce the inference loop to conceptualize how concept and construct validity relate to the wider scientific process (Figure 1). An inference loop is defined as the iterative process of repeatedly adjusting a concept, construct, analysis, or finding following new evidence and theorizing. The goal of an inference loop is to build intersubjective agreement on the validity of knowledge produced at different steps in the scientific process. It is inspired by the back-propagation algorithm that enables machine learning models to learn by updating their weights based on the results of a training iteration (Rumelhart et al., 1986). Such back-propagation provides a powerful analogy for the repeated adjustment process that enables learning (Lillicrap et al., 2020), not just at an individual level, but also in science.

A concept is only valid if it is clearly delineated from other concepts, robustly defined in theory, and operationalized in empirical research (Bringmann et al., 2022; Locke, 2012). Importantly, new evidence can always emerge to undermine concept validity (e.g., failure to replicate and operationalization challenges). The same applies for

Figure 1

Locating Construct and Concept Validity Within the Research Process



Note. Solid lines denote the conventional linear scientific process. Dashed lines denote the repeated backpropagation processes of validation that can result in revisions of prior stages. Our focus is on the concept and construct validity processes (shaded in black).

constructs (Flake & Fried, 2020), for empirical studies (Banks et al., 2016), and scientific findings (Freeman, 1973). Thus, inference loops power every step of the scientific process, allowing for the backward propagation of new evidence to adjust any of the previous steps. Through the lens of inference loops, validity is not a static state but a dynamic process for maximizing intersubjective agreement on utility (i.e., what works; Gillespie et al., 2024).

Despite its importance, validity is rarely reported in psychological studies (Flake et al., 2022). This issue extends to quantitative text analysis, where data science has generally prioritized accurate predictions over measurement validity (Boyd & Schwartz, 2021; Song et al., 2020). This is problematic because texts are an underutilized and rich source of psychological data (Jackson et al., 2022). Texts are generally unobtrusive, meaning they occur naturally without interference from the researcher (Webb et al., 1966). This instills them with ecological validity, meaning they reflect how people act and behave outside of laboratory conditions (Albert & de Ruiter, 2018; Andersen, 2025).

Texts are increasingly analyzed using classifiers, which computationally assign a category (or categories) to a text. Classifiers are either deductive or inductive, where the former involves sorting texts into "known categories," and the latter involves the generation of "unknown categories" through clustering texts (Grimmer & Stewart, 2013, p. 268). The focus of our contribution is on deductive classifiers because they aim to measure preexisting concepts, making them comparable to more traditional measurement instruments (e.g., surveys).

Rule-Based Classifiers

Rule-based classifiers employ a set of researcher-defined rules for assigning categories to texts. The rule-based dictionary method was the first text classification approach employed in psychology (Boyd & Schwartz, 2021; Stone et al., 1966). It involves using word-counting rules for representing a concept (if term X occurs, add 1 to the count). To illustrate, the words "happy" and "joy" might imply the presence of positive emotions in a text, while "sad" and "disgust" might imply negative emotions. These dictionaries are created manually.

Rule-based dictionary classifiers became the conventional method for quantifying text in psychology with the advent of the Linguistic Inquiry and Word Count (LIWC) tool (Pennebaker & Francis, 1999). LIWC's ability to be used "off-the-shelf" made it a go-to tool for psychological and social scientific research (Grimmer & Stewart, 2013). It formalized word counting, making it easy to use, and provided a large number of validated dictionaries (Boyd et al., 2022; Pennebaker et al., 2015). Rule-based classifiers are perfectly reliable in terms of reproducibility because they always generate identical results if applied to the same data. This means measurements can easily be replicated. Rule-based classifiers are also transparent because one can observe how the rules are applied.

Compared to other rule-based classifiers, the LIWC dictionaries benefit from ongoing evaluation and updating using psychometric conventions (Boyd & Schwartz, 2021; Tausczik & Pennebaker, 2010). Researchers have extensively studied the predictive capabilities of LIWC. For instance, meta-analyses have shown systematic correlations between LIWC dictionaries and various personality scales (Chen et al., 2020; Holtzman et al., 2019; Tackman et al., 2019). These studies showed that the LIWC dictionaries have some construct validity for measuring personality concepts. However, these studies also highlighted how this construct validity has an upper bound because rule-based classifiers struggle with measuring complex latent concepts that are not manifest in simple word counts. To illustrate, the meta-analyses employed LIWC dictionaries to predict personality rather than measuring it directly. The effect sizes of the meta-analyses were small (all $r < .10$ or $|r| < .10$), highlighting the limited construct validity of using LIWC dictionaries for measuring personality in text (Chen et al., 2020).

Rule-based classifiers' main drawback is that they do not capture the contextual meaning of words. Dictionaries represent a text by a decontextualized "bag of words" that does not represent grammar or word order (Grimmer & Stewart, 2013; Qader et al., 2019). Although more complex rules are possible, dictionaries typically count the frequency of words and ignore context (e.g., "not happy" would still count as happy). Speech acts (word, utterance, book, etc.) can mean very different things in different contexts (van Dijk, 1977). Accordingly, dictionaries are limited by their perfect reliability; because dictionaries are fixed, they cannot adapt the meaning of words to different contexts.

Supervised Machine Learning Classifiers

Supervised machine learning classifiers provide an alternative to rule-based classifiers for measuring psychological concepts in text. Machine learning is defined as "computer systems that automatically improve their performance through experience" (Mitchell et al., 1990, p. 417). Consistent with known and unknown classifier types (Grimmer & Stewart, 2013), machine learning classifiers are either "supervised," which focus on deductive measurement, or "unsupervised," which focus on inductive clustering (Eichstaedt et al., 2021). We only focus on supervised classifiers because they are deductive.

Supervised classifiers categorize texts through a statistical model that is trained on data coded for the target categories. These data have generally been produced through manually coding texts for concepts (Song et al., 2020). Modern supervised classifiers employ deep learning, which leverages artificial neural networks to represent the input data (Urban & Gates, 2021). These networks are modeled on the processes of representation found in biological neurons in human and animal brains (Goodfellow et al., 2016).

Modern supervised classifiers quantify texts using dense vectors, trained to understand the contextual meaning of words (Chernyavskiy et al., 2021; Jackson et al., 2022; Wolf et al., 2020). Each word in each text is represented by a vector denoting a position in a high-dimensional feature space. This vector encodes word-meaning by examining the words that occur nearby the target word. In practical terms, the phrases "I'm not happy" and "I'm happy" will be represented differently by the model. Supervised classifiers can therefore measure complex concepts more directly than rule-based classifiers and, consequently, offer increased construct validity.

Supervised classifiers also benefit from a simple evaluation process, inherited from data science. First, the manually coded data set is split into training and test data. The training data are used for developing and refining the algorithm, while the test data are withheld until training is completed and used to evaluate the algorithm. Evaluating the final model on withheld test data provides evidence of construct validity because accurate predictions on unseen data indicate the algorithm has generalized the categories from the training data.

Withholding test data enables an inference loop for classifier development. Researchers can safely repeatedly adjust the training

process, fine-tuning model parameters and strategies until they are ready to evaluate the model on the test data. Data splitting is also essential to the algorithm's training process, which works as an automatic inference loop. A subset of the data (the validation data) is separated from the training data and used to measure prediction errors, update internal weights, and guide hyperparameter tuning after each training iteration. By monitoring performance on validation data, the researcher determines when to conclude training and carry out the final evaluation on the test data.

The withholding of test data is also an essential feature of the common task framework, which is the "secret sauce" behind data science's exponential growth (Donoho, 2017, p. 752). First, a coded data set is made publicly accessible with a competition to build the best supervised classifier. Researchers then build a model to generate the best evaluation metrics, entering the competition by sharing their final protocol publicly. When the competition ends, a referee assesses the best classifier using withheld test data. Through its inference loop structuring, the common task framework has driven significant advances in machine learning models, highlighting the efficacy of withholding test data to develop measures.

The main drawback of supervised classifiers is that their construct validity is dependent on the construct and concept validity of the manually coded data used for training and testing. If the manually coded data lack validity, so will any text classifier trained on it, regardless of its performance on the test data (garbage in, garbage out). As expressed by Song and colleagues, "there is a substantially greater risk of a researcher reaching an incorrect conclusion regarding the performance of automated procedures when the quality of manual annotations used for validation is not properly ensured" (2020, p. 550). This foundational manual coding stage is rarely scrutinized by data scientists, and validity has often been overlooked when quantifying text (Birkenmaier et al., 2024). Instead, their focus is generally on developing new model architectures for improving predictions (e.g., Chernyavskiy et al., 2021), and prediction is possible without validity (Anderson, 2008).

LLM Classifiers

LLMs offer a new way of classifying texts. LLMs are deep learning models trained on huge quantities of textual data (Silva & Hassani, 2023). Modern LLMs are generative in that they output content (e.g., text, photos, and videos) in response to a user prompt, generally written in natural language. Prompts can be used to direct an LLM on how to classify text data (e.g., "Is the following text happy?"; Bahrami et al., 2023). The LLM's response (e.g., "Yes") is taken as the classifier's prediction. LLM classifiers are characterized by how many examples of correct classifications are included within the prompt. An LLM classifier is zero-shot when it includes no examples, one-shot when it includes one example, and few-shot when it includes multiple examples.

The main benefit of LLM classifiers is their ease of use and accessibility. Rule-based classifiers are time-consuming to make, and supervised classifiers require large amounts of training data, even in cases of fine-tuning a pretrained algorithm (Sun et al., 2019). In contrast, LLM classifiers can be developed without any training data because they rely on prompts. Developing an LLM classifier can be performed exclusively through iterative prompt engineering without hyperparameter sweeps. This enables quick and efficient adjustments to the LLM classifier compared with supervised classifiers, which require time and computationally intensive training or finetuning. The use of natural language to prompt LLM classifiers also makes machine learning methods more accessible to researchers with fewer programming skills. LLM classifiers require some coding knowledge to implement (e.g., for automated application of the prompt); however, this is minimal compared to supervised classification.

Another benefit of LLM classifiers is that they enable an ongoing assessment of content validity during their development through an inference loop. Content validity is defined as the degree to which the content of a construct (e.g., a survey item) reflects the underlying concept being measured (Cronbach & Meehl, 1955). Content validity cannot be adjusted during the development of supervised classifiers, which rely on post hoc analysis of how the algorithm is making its classification decisions (Ribeiro et al., 2016). LLM classifiers resemble rule-based classifiers in that the researcher adjusts natural language parameters while the language model itself is held constant. In contrast, supervised classifiers update their model through learning the input data.

The main drawback of LLM classifiers is that their concept and construct validity depend on manually coded data; the same problem is associated with supervised classifiers. Another drawback is that LLMs are stochastic parrots, meaning they respond to the prompt probabilistically without understanding the meaning or significance of the interaction (Bender et al., 2021). LLM classifiers can produce different results based on the same input data, making them less reliable than rule-based or supervised classification. The large amounts of data required for creating an LLM means it has likely been trained on dubious texts with undesirable content (e.g., racial bias and sexism). This risks a construct validity problem where the prompt may not be understood in the way the researcher intends it to. Finally, LLMs are also novel, meaning their viability for psychological research has largely gone untested (Demszky et al., 2023). That said, early research shows significant potential for integrating LLMs into psychological and social scientific research (Rathje et al., 2023; Ziems et al., 2023).

Integrating Manual Coding Into Psychological Text Classifier Development

Construct validity is established for supervised and LLM classifiers by assessing their predictions against withheld manually coded test data. This evaluation method enables direct comparison between different types of classifiers (e.g., rule-based, supervised, and LLM) by assessing their performance on the same test data. As long as classifiers are developed to produce the same predictions (e.g., a binary categorization), their performance can be compared using evaluation metrics (van Atteveldt et al., 2021). However, these metrics are only useful if the manually coded data are valid to begin with (Song et al., 2020).

Content analysis (Krippendorff, 2019) is the conventional approach for rigorous manual coding. Content analysis has been used extensively in psychology and the broader social sciences (Neuendorf, 2017). For instance, the method has been employed for measuring emotions (Shiraishi & Reilly, 2022), workplace personality (Ragsdale et al., 2013), and children's happiness (Park & Peterson, 2006). Content analysis is deductive in that it defines the constructs prior to operationalization (Neuendorf, 2017). It establishes rigor by having two or more human coders score the same data and quantifying their interrater reliability (Hayes & Krippendorff, 2007). Interrater reliability is a minimum requirement for construct validity (rather

than a direct measure) because, without it, the coders are coding different underlying concepts (Krippendorff, 1970).

Despite the rigorous conventions, content analyses can still suffer from concept validity issues, regardless of interrater reliability between coders and construct validity checks. If a concept is poorly theorized, coders can still produce reliable coding in the same way text classifiers can accurately predict low-validity manual coding. To illustrate, there exist multiple codebooks for assessing the deliberative quality of dialogue (Friess & Eilders, 2015); however, these have often measured the same concept with different definitions (Beauchamp, 2020; Goddard & Gillespie, 2023). Consequently, the underlying concept being measured may differ significantly between studies, indicating concept and construct validity problems.

Psychologists have recommended using an abductive approach to identify and address concept validity problems (Gillespie et al., 2024; Muthukrishna & Henrich, 2019). Abduction is a third and underutilized form of inference that focuses on generating new theory by identifying and explaining surprising findings (Peirce, 1965). Deduction is used to test theories, induction is used to organize observations, and abduction is used to generate new theory by explaining anomalies. For instance, given the theory that all mammals have hair, it follows that all dogs should also have hair (deduction). However, if a dog is found to have no hair, an alternative theory is required to explain this anomaly; the abduction might be that the dog is sick. Without the ability to use alternative theories to explain surprising results, science would not be able to progress, because it would forever be stuck in a single explanatory paradigm (e.g., dogs are not mammals if they have no hair). Abduction is related to inference loops as researchers may have to move beyond existing explanations for measurement problems (e.g., a poor model) to explain validity problems (e.g., conceptual incoherence).

This article addresses validity problems in text classification with the RAMP method. The method integrates psychometric validation using abduction with best practices for manual coding (content analysis) and classifier development (data science). It uses the concept of inference loops to integrate the three approaches into a single method, aimed at maximizing validity both within and across studies. Because inference loops transcend the different literatures and practices, they provide a robust and flexible framework for psychologists to use when navigating the rapidly evolving field of text classification. The RAMP method also contributes to data science by conceptualizing a method for improving the validity of manual coding and, subsequently, classifier development.

The RAMP Method

The RAMP method has three stages (Figure 2): a manual coding stage for creating a data set, a classifier development stage that uses this data for creating one or more text classifiers, and an integrative evaluation stage for making a global assessment of concept and construct validity. Each stage of RAMP is powered by an inference loop.

The first stage of RAMP (manual coding) employs an inference loop to maximize the construct validity of the manual coding. The loop involves human coders scoring a sample of the raw data (calibration data), evaluating the reliability of their coding, and evaluating problems (e.g., discrepancies in coder's interpretations of the concept) to make informed decisions on how the codebook should be adjusted. The loop ends when predefined goals have been met (e.g., high interrater reliability and saturation of operationalizations). The final codebook is subsequently used to score the full data set.

The second stage of RAMP (classifier development) employs an inference loop to create one or more text classifiers using the manually coded data from the first stage. Before conducting the loop, a portion of the manually coded data (test data) is withheld for calculating final evaluation metrics. The loop involves training or programming a classifier (the protocol), calculating evaluation metrics on validation data, evaluating problems in classification (e.g., misclassifications and low metrics), and using these insights to adjust the protocol. As with the manual coding, the loop ends when a set of goals are met (e.g., reaching an upper bound on evaluation metrics). The final classifier is then applied to the test data to calculate evaluation metrics for reporting.

The third stage of RAMP (integrative evaluation) employs an inference loop to make a global assessment of the construct and concept validity of the manual coding and text classifier. The loop involves explaining surprising findings (e.g., interrater discrepancies and classifier misclassifications) from the two previous stages in the context of guiding theory, questioning whether the explanation adequately explains the surprises, and adjusting the explanation accordingly. The loop ends when the researcher is satisfied with their explanation and makes an integrated assessment of the manual coding and text classifier's construct and concept validity. The integrative evaluation stage builds on an ongoing assessment of concept and construct validity (dashed arrows, Figure 2), integrating any previous findings into a final evaluation of validity.

By employing abduction, the integrative evaluation stage is geared toward concept validity because it requires the researcher to make sense of outliers using theoretical and empirical insights gained during manual coding and classifier development. The detection of concept validity problems is prioritized because conceptual issues can invalidate any presumed construct validity established during previous stages (Figure 1). To illustrate, it could be uncovered that a classifier with high evaluation metrics for measuring a novel personality trait—developed on manually coded data with high interrater reliability—is measuring a preexisting personality trait (e.g., through assessment of correlations with other measures) or is incoherent and includes clearly distinct phenomena (e.g., through qualitative assessment of the classifications). In these cases, the classifier is either not measuring the intended concept (construct validity problem) or the concept itself is confused (concept validity problem).

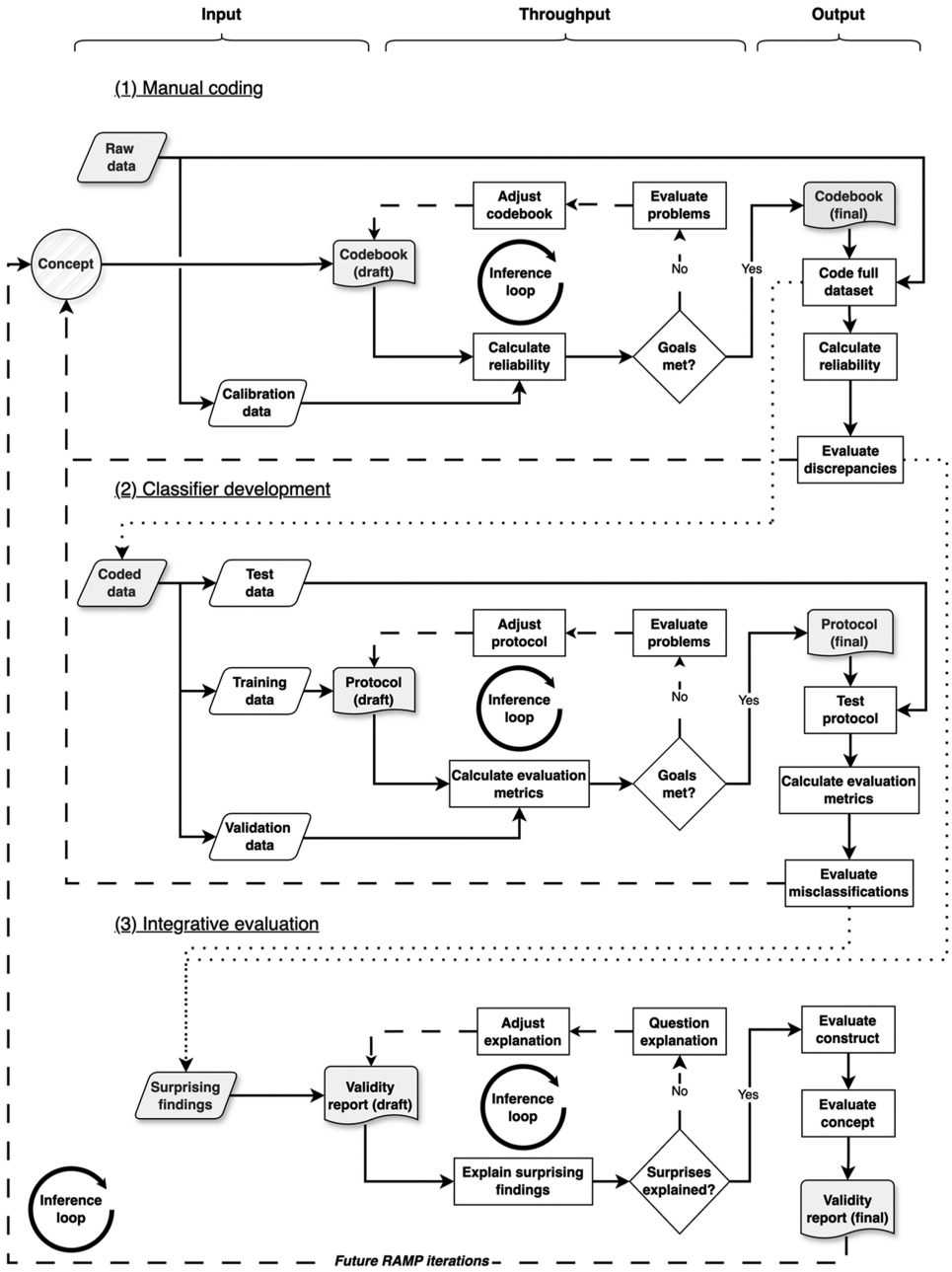
Figure 2 also includes a fourth inference loop to indicate that RAMP is part of an ongoing effort of adjusting concepts and theories between studies. This incorporates a procedural definition of validity as the accumulation of intersubjective agreement on the utility of a concept and measurement tool. RAMP emphasizes incremental validity (Y. Feng & Hancock, 2022), meaning it works to alter protocols and concepts iteratively based on repeated attempts at manual and automated operationalization. The method is novel because it promotes an iterative evaluation of validity within stages, between stages, and across studies. These iterative processes span classifier development processes that have hitherto been conceptualized separately. The next sections describe the three stages in detail.

Manual Coding Stage

The first stage leverages an inference loop for creating a rigorous manually coded data set. The input phase of manual coding requires

Figure 2

Process Diagram for the RAMP Method



Note. Parallelograms indicate data sets, rectangles indicate processes, wave rectangles indicate documents, and diamonds indicate decisions. Grayed objects denote inputs and outputs. Solid lines show the development (forward) processes, dashed arrows show evaluation (backward) processes, and dotted lines represent dependencies between RAMP stages. RAMP = Repeated Adjustment of Measurement Protocols.

a draft codebook for adjustment in the inference loop. Following content analysis conventions, the draft codebook should take into account previous operationalizations of the concept and be justified in existing theory (Neuendorf, 2017). The throughput phase contains the inference loop where coders repeatedly apply the latest version of the codebook to a sample of the full data set (calibration data). Each application is followed by an assessment of interrater reliability and interrater discrepancies to infer construct validity problems and, subsequently, adjust the codebook. Ending the throughput phase requires evaluating whether the researcher's goals have been met. Goals involve stopping rules, which should include saturation of textual instances of the target phenomena, reaching adequate reliability, and any relevant practical considerations (e.g., coder fatigue and limited budget).

A similar iterative process of codebook development is common in manual coding (e.g., MacPhail et al., 2016). For instance, Krippendorff states that content analysis "may include iterative loops—the repetition of particular processes until a certain quality is achieved" (2019, p. 85). In practice, however, codebooks are often finalized before coder training and empirical testing (Neuendorf, 2017, 2018). In contrast, RAMP's manual coding stage makes visible how coder training is used to develop the final codebook through repeated adjustment. This iterative process, we argue, is central to improving validity.

Deliberations with coders can help identify validity problems because they encourage intersubjective agreement on the causes of discrepancies in their coding. For instance, coders may not fully understand how to apply the codebook (construct validity), be unclear on how the operationalizations represent concepts (construct validity), or have trouble understanding the conceptualizations themselves (concept validity). RAMP employs deliberations to uncover validity problems, as these provide an environment for abductive insights that can feedback into the concept and construct to improve validity (Krippendorff, 2019; Timmermans & Tavory, 2022).

The output phase involves manually coding the full data set using the final codebook. The data are split among coders, and interrater reliability is calculated on a shared subset of the coded data for reporting. The final coding is done blind, with coders unaware of which texts belong to this subsample. The choice of interrater reliability statistic depends on the type of variable (binary/categorical, ordinal, continuous), the number of coders (two or more), and whether the coded data are imbalanced (Gisev et al., 2013). Imbalanced data create problems for reliability as coders can achieve high values by attributing everything to the same category. Table 1 provides an overview of common statistics and when they should be used (G. C. Feng, 2014).

Reliability statistics generally produce a value between 0 and 1. Ideally, researchers should aim for the highest reliability possible. Content analysis recommends that a Krippendorff's α should be .90 or above, but values between .80 and .90 are considered acceptable (Krippendorff, 2019; Neuendorf, 2017). In practice, however, this may not be achievable due to coder fatigue, variability in coders' interpretation of the codebook, and validity issues unknown prior to coding (Krippendorff, 2019).

Interrater reliability is the primary statistic used in content analysis as it is a prerequisite for validity. But interrater reliability is not validity. To illustrate, coders could score texts describing dogs under a "cat" category (high reliability), but this does not mean the texts are describing cats (low validity). However, if coders cannot agree on what texts should be coded under the dog category (low reliability), the scores are inherently invalid. Thus, establishing validity not only requires reliable coding but also conceptual assessments during codebook development.

The output phase may also include further validity checks, such as having coders score for similar concepts and then assess the results against codes for the target concepts (discriminant validity). RAMP is flexible in allowing different development pipelines; however, more evidence for validity is always better for identifying problems and addressing them as early in the pipeline as possible.

The RAMP method's manual coding stage integrates content analysis (Krippendorff, 2019) with data science principles through the overarching conceptualization of inference loops. Data science best practices emphasize iterative development of accurate measurement tools through the withholding of test data and a common task framework (Donoho, 2017). Content analysis best practices emphasize iterative practice in the conceptualization of the codebook prior to coder training (Krippendorff, 2019; Neuendorf, 2018). RAMP integrates both through the concept of inference loops, aiming to maximize validity.

Classifier Development Stage

The second stage employs another inference loop to develop a classifier using the manually coded data produced in the first stage. The input phase requires choosing the type of classifier to develop and the splitting of the manually coded data into training and test data. For supervised machine learning, the split sizes are generally weighted toward the training data (60%–80%) with the remaining (40%–20%) used for the test data (Rosenbusch et al., 2021). However, rule-based and LLM classifiers do not rely on training data and, therefore, can use a larger proportion of the data set as test data. By withholding the test data from the throughput phase, it can be used to compare multiple classifiers. In this case, the choice of split size is determined by the classifier selection (e.g., supervised classifiers require most training data) and the test data must be the same for all classifiers.

The throughput phase is structured by an inference loop for each classifier being developed. At a macro level, each classifier is developed by creating the protocol (e.g., choosing terms, training a classifier, and writing a prompt), calculating evaluation metrics on validation data, evaluating problems in its performance, adjusting the protocol, and repeating the process. Evaluation metrics are determined by the number of correct and incorrect classifications. As with interrater reliability, the choice of metric should be informed by the

Table 1
Interrater Reliability Statistics

Statistic	Variable type	Number of coders	Appropriate for imbalanced data
Cohen's κ	Binary, categorical	Two	No
Spearman ρ	Ordinal	Two	Yes
Krippendorff's α	Binary, continuous, categorical, ordinal	Two or more	Partial
Gwet's AC1	Binary, categorical	Two or more	Yes
Gwet's AC2	Continuous, ordinal	Two or more	Yes
ICC (1)	Continuous	Two or more	No
ICC (2)	Continuous	Two or more	No

Note. AC1 = agreement coefficient 1; AC2 = agreement coefficient 2; ICC = intraclass coefficient.

skew of the data as high values can be misleading if the classifier assigns predominantly a single category. The simplest evaluation metrics are those used in binary classification, where misclassifications are organized as true or false positives or negatives. Four common statistics are the accuracy of the binary classifier (ratio of correct classifications to the total number of classifications), the precision (ratio of true positives to predicted positive classifications), the recall (the ratio of true positives to correct classifications), and the harmonic mean of the precision and recall (F1 score).

The four basic evaluation metrics can reveal different construct validity problems. The accuracy statistic is the least informative but is useful for a rough gauging of a classifier's overall performance. Precision reflects the accuracy of positive classifications, where low values indicate a high number of false positives. Recall reflects the classifier's ability to identify positive cases, where low values indicate a high number of false negatives. The F1 score is the harmonic mean of the precision and recall and is functionally similar to accuracy. These metrics can be used to steer adjustments to a classifier. Accuracy and F1 scores give an overall indication of performance, precision problems indicate the classifier is too broad in its representation of the construct (overfitting), and recall problems indicate that it is too narrow (underfitting).

Rule-based, supervised, and LLM classifiers have different parameters that can be adjusted during the inference loop. They should therefore be developed in conjunction with existing literature and conventions. A rule-based protocol consists of, at minimum, a list of terms and a scoring method to be adjusted through the inference loop. Terms can be added or removed based on evaluation metrics and misclassifications. For instance, adding terms may improve recall but lower precision, meaning the number of false positives has increased. Examining false positives can thus help determine which terms are causing precision problems and guide their removal in the next iteration. Altering the counting method can also affect the evaluation metrics. For instance, a classifier could generate many false positives if it were designed to classify a text when identifying the presence of a single term (positive if text contains "happy" or "joy"). Increasing the threshold to two or more words (positive if text contains "happy" and "joy") might, therefore, reduce the false positive rate.

A supervised protocol requires, at minimum, a splitting of training and validation data, selecting a model (e.g., Bidirectional Encoder Representations From Transformers [BERT]), and specifying appropriate hyperparameters, which steer the model's training. During the inference loop, the model is trained and subsequently evaluated on the validation data. Unlike rule-based and LLM classifiers, a supervised classifier cannot be adjusted using insights gained from examining misclassifications because the model is determined automatically through its training process. Because a new model is trained after each inference loop, the evaluation metrics (calculated on the validation data) are used to steer changes in the hyperparameters. For instance, increasing one parameter might improve the overall performance of the classifier, but increasing another might reduce performance. Once hyperparameters stabilize, the researcher may try different training and validation splits to see if they consistently lead to improved performance across different combinations of data.

An LLM protocol requires an initial prompt and a chosen generative model (e.g., GPT-4, Claude, and Gemini); it may also include numerical parameters (e.g., temperature to change the variability in the output text). As with rule-based classifiers, the evaluation metrics and misclassifications are used to steer adjustments. For instance, a change in the prompt may lower recall, indicating the change has increased the number of false negatives. Examining the misclassifications can be used to determine which types of texts the classifier is failing to identify, and the prompt can be broadened accordingly (e.g., adding more example cases and extending the concept definition). Similarly, a change of model might improve all evaluation metrics, showing that the new model is better at identifying the target concept.

For any classifier, the decision to end the inference loop and proceed to the output phase is informed by stopping rules. These can involve a mix of goals relating to evaluation metrics (e.g., improved performance over preexisting classifiers and plateauing in a set number of inference loops), protocol features (e.g., adjusted all relevant hyperparameters and trialed a set number of models), practical components (e.g., limited resources), and qualitative assessments (e.g., saturation on LLM prompt or rule-based terms). We should also note that stopping rules can be overridden if there is justification to do so. For instance, the first iteration may have improved evaluation metrics over preexisting classifiers, or the researcher might decide to trial a new model. RAMP enables dynamic updating to maximize construct validity in the throughput phase, including the stopping rules themselves.

In the output phase, the researcher evaluates the best classifier from the throughput phase on the withheld test data. It is essential that the test data are kept independent from the training (and validation) data to ensure that the developed classifier can generalize to new data and that different classifier types can be compared against each other. If calculated on withheld test data, the evaluation metrics reflect a degree of construct validity so long as the manual coding and concepts measured are valid. This phase can also include other validity checks, such as whether classifier predictions correlate with similar measures (concurrent validity), predicts real-world behaviors (ecological validity), or can be applied in other contexts (transfer validity).

Integrative Evaluation Stage

The goal of the integrative evaluation stage is to uncover and explain problems across manual coding and classifier development stages to make an integrated assessment of concept and construct validity. This stage emerged from the literature on psychometric validation using abduction (Gillespie et al., 2024), and is not a typical component of the text classification development pipeline. In the input phase, surprising findings from the previous stages are collated and given an initial explanation. A surprising finding refers to any observation during manual coding or classifier development that required explanation (Peirce, 1965). This might include any unexpected adjustments to the codebook, unresolved problems with edge cases in the coding manual, aspects not captured, or unexpected problems with the performance of the classifier.

The throughput phase involves brainstorming, deliberating, and potentially post hoc manual coding or statistical analysis to evaluate possible explanations until a satisfactory one is found. Identifying explanations may not require abduction (e.g., if it emerges from random error) and may not yield any concept validity issues. However, an inference loop should be conducted to make an integrated assessment on whether the cause of the surprising findings is conceptual. No psychological measure is perfect, and the development process

may have obscured conceptual problems through high reliability or evaluation metrics.

The output phase reports the explanations for the surprising findings in relation to concept and construct validity problems. The integrative evaluation stage aims to identify theoretical and conceptual anomalies and try to resolve them through an abductive approach. It emphasizes concept validity and, therefore, speaks to current issues in psychological theory and methods (Eronen & Bringmann, 2021; Muthukrishna & Henrich, 2019). Abductive processes are essential to validating manual coding in content analysis, and "a computer aided content analysis should do the same" (Krippendorff, 2019, pp. 260–261) for determining validity. This justifies RAMP's integration of the manual coding and classifier development stages for evaluating validity, as both stages can produce surprises that can feedback into the current or prior stage to increase validity. The output of the integrative evaluation stage requires transparent reporting of concept and construct validity problems as to aid subsequent researchers in the macro between-studies inference loop. These reports can build off each other and be used to incrementally increase the validity of both our concepts and the way we measure them in text.

Case Study

We assessed the viability of RAMP through an empirical case study. The study compared RAMP with a noniterative method for developing text classifiers for measuring misunderstandings in online dialogue. We defined a noniterative method as the absence of repeated adjustments (i.e., inference loops) in generating the manually coded data set and developing classifiers. Noniterative methods are rarely done in practice; however, by using them as a baseline, we could investigate the added value of the repeated adjustments recommended by the RAMP method.

We chose to measure misunderstandings in online dialogue for two reasons. First, the theoretical role of misunderstandings in online dialogue quality is unclear. On the one hand, misunderstandings might drive deliberation by highlighting problems in mutual understanding (Stromer-Galley, 2007). On the other, misunderstandings might drive incivility as individuals become frustrated from feeling misunderstood. Second, psychology has mainly used surveys (e.g., Lees & Cikara, 2020; Livingstone et al., 2020; Rubin, 1994) or mixed experimental-qualitative designs (e.g., Corti & Gillespie, 2016; Heasman & Gillespie, 2018) to study misunderstandings, with no text classifier available to measure the phenomenon in context.

Misunderstanding is defined as an individual recognizing a problem in mutual understanding during dialogue (Table 2). Thus, misunderstanding is produced in social interactions by the combination of different levels of perspective taking between a self and other (Laing et al., 1966). Comparing direct perspectives reveals agreement (I think X and you think X), comparing direct and metaperspectives reveals understanding (you think X and I think you think X), and, finally, comparing direct perspectives and meta-meta-perspectives reveals felt understanding (I think X and I think you think I think X). For example, uttering the phrase "you don't understand my point" indicates the self's meta-meta-perspective does not match with the self's actual perspective.

The case study assumed that misunderstanding had a metaperspective and meta-meta-perspective component. It employed an initial (conceptual) definition of misunderstanding as any instance in an online dialogue where the self directly misunderstands the other or feels misunderstood. As misunderstandings are internal states, we operationalized the concept through reported misunderstanding, where the self makes explicit either direct or felt misunderstanding in their contributions to the dialogue.

Method

Data

The data set comprised sentences from online dialogues, referring to a linear thread of comments, organized by a "reply-to" function. We sampled dialogues from three sources: Reddit data downloaded using the website's Application Programming Interface; "Twitter Customer Support" data (Thought Vector & Axelbrooke, 2017), involving dialogues between organizations and Twitter users; and Wikipedia Talk Pages data (Danescu-Niculescu-Mizil et al., 2012), involving dialogues between editors about the content of Wikipedia articles. The three sources were chosen to represent different public communication contexts. About 1,000 dialogues were randomly sampled from each data set, and 230 were removed manually for being unusable (e.g., not in English), yielding 21,884 sentences from 2,770 online dialogues. Table 3 reports the distribution of authors, dialogues, and sentences across the three data sources. The same data set was used for both noniterative and RAMP methods.

Sentences were used as the unit of analysis, and misunderstandings were coded as a binary variable, determined by the presence of either direct or felt misunderstandings (Table 2; Supplementary Materials A and B in the online supplemental materials). The manual coding thus involved coders scoring sentences as misunderstandings, and classifiers were used to reproduce this binary categorization. Sentences were parsed using the spaCy (Honnibal et al., 2022) Python package and anonymized by algorithmically replacing usernames, locations, hyperlinks, and dates with fake data. The study and Reddit data collection were approved by the host institution's Research Ethics Committee (Reference: 56581).

Table 2
Perspective Taking and Misunderstanding According to Laing et al. (1966)

Concept	Self	Other	Matching
Agreement	Direct perspective, I think X	Direct perspective, I think X	Yes
Disagreement	Direct perspective, I think X	Direct perspective, I think <u>not X</u>	No
Understanding	Direct perspective, I think X	Metaperspective, I think you think X	Yes
Misunderstanding	Direct perspective, I think X	Metaperspective, I think you think <u>not X</u>	No
Felt understanding	Meta-meta-perspective, I think you think I think X	Metaperspective, I think you think X	Yes
Felt misunderstanding	Meta-meta-perspective, You think I think <u>not X,</u> but I think X	Metaperspective, I think you think <u>not X</u>	No

Note. Underlined negations for emphasis.

Table 3
Case Study Data Set Distribution

Source	*N* authors	*N* dialogues	*N* sentences
Reddit	1,833 (43%)	899 (32%)	7,884 (36%)
Twitter Customer Support	1,081 (25%)	921 (33%)	5,621 (27%)
Wikipedia Talk Pages	1,383 (32%)	950 (34%)	8,489 (39%)
Full data set	4,297	2,770	21,884

Applying the RAMP Method

For the manual coding stage, the study employed five coders of MSc level or above in psychology to classify the data. In the input phase, we defined a draft codebook and gathered the raw data. In the throughput phase, we used a small sample of dialogues ($n < 100$) to test the coders' interrater reliability for each new version of the codebook. In the throughput, we used Krippendorff's α (1970) to quantify interrater reliability as it is a flexible reliability measure and fairly robust to skewed distributions (Hayes & Krippendorff, 2007). For the output phase, and once the distribution of the data was revealed, we also calculated Gwet's agreement coefficient 1 (AC1) as it is recommended for highly imbalanced data (G. C. Feng, 2014). To troubleshoot the results, we deliberated with coders about the scoring procedure in the context of the literature and the latest reliability statistics and coding discrepancies. We conducted five inference loops, stopping when there was alignment between coders on the final codebook and acceptable interrater reliability.

In the output phase, the coders scored the full data set. Each coder received between 6,400 and 6,700 sentences to code, 1,610 of which were common to all coders (the numbers varied as sentences were embedded in dialogues with different lengths). The shared subset was used for calculating the interrater reliability of coders. When building the final data set, scoring discrepancies were resolved by following the majority consensus, where three or more coders had to agree it was or was not a misunderstanding.

For the classifier development stage, we created rule-based, supervised, and LLM text classifiers using the manually coded data produced in the previous stage. All three classifiers therefore had the same unit of analysis (sentences scored for misunderstandings as a binary variable). For the input phase, we split the coded data into training/validation (70%) and test (30%) data. For the throughput phase, we used the training and validation data to conduct an inference loop for each classifier. For the output phase, we compared the best throughput classifiers on the withheld test data using a variety of evaluation metrics. We calculated the accuracy and the weighted F1 score as these are commonly used evaluation metrics for binary classification (Hand & Christen, 2018). Because our data were heavily skewed, we also calculated metrics appropriate for imbalanced data: the Matthews correlation coefficient (MCC; Chicco & Jurman, 2020), the balanced accuracy, and the F1 score, precision, and recall for the positive category (misunderstandings). These metrics also guided the classifier adjustments in the throughput phase.

For all three classifiers, we stopped the iterations when we observed a plateauing or decreasing of evaluation metrics from changes to the classifier. For the purposes of demonstration and comparison, we conducted the same number of inference loops (21) for each classifier. This number was determined by the number of iterations conducted in the first classifier developed (rule-based). We then proceeded to assess each classifier on the test data.

Each classifier required a different inference loop. For the rule-based classifier, we adjusted terms and term type (words and word sequences) across iterations. We kept the scoring method the same, where we assumed a sentence indicated misunderstanding if any term was present. A ratio was inappropriate because sentences are short, meaning the presence of multiple terms was unlikely. The classifier was developed by inputting a set of terms, applying them to the training data, troubleshooting results, adjusting the terms, and repeating until satisfied that no more terms could be added or removed. The troubleshooting was performed by examining misclassifications to remove or adjust terms, brainstorming new terms, and deliberation among authors.

For the supervised classifier, we fine-tuned a BERT (Devlin et al., 2019) model for each inference loop iteration. This entailed adding an additional classification layer on top of BERT's existing 12 layers of transformers. We chose BERT because it performs well on small data sets (González-Carvajal & Garrido-Merchán, 2021) and has been previously used for psychological text classification (Biggiogera et al., 2021; Jun et al., 2021; Kumar & Jain, 2022). We adjusted the classifier by changing hyperparameters after calculating evaluation metrics on validation data (sampled from the training data). We varied the proportion and sampling of data used for validation (between 10% and 30%) after each training cycle was completed. This helped determine whether the hyperparameter changes led to consistent improvements to the evaluation metrics across different data splits.

We focused on three hyperparameters—epochs, batch size, and learning rate—chosen based on recommendations from the literature (Devlin et al., 2019; Sun et al., 2019). The epochs are the number of times the algorithm iterates over the training data to estimate its parameters (Urban & Gates, 2021, p. 16). The batch size is the number of data points observed by the algorithm before it refines its parameters during training (Smith, 2018). Third, the learning rate is the size of the steps taken by the algorithm for estimating the parameters' optimal values through gradient descent (Goodfellow et al., 2016, pp. 81–82). BERT has been found to work best using two, three, or four epochs, a batch size of 16 or 32 samples, and learning rates of $5e-5$, $3e-5$, or $2e-5$ (Devlin et al., 2019). We cycled through these different parameters through the 21 inference loops to determine the optimal configuration for the final classifier.

For the LLM classifier, we used OpenAI's GPT-4o model (version from May 13, 2024) and adjusted the prompt. We chose a GPT-4 model (OpenAI et al., 2024) as it performs better on most tasks than open-source alternatives (Nadeau et al., 2024). As with the rule-based classifier, misclassifications were used to determine areas of improvement for the prompt following an assessment of evaluation metrics calculated on a different random subset of the training data for each iteration. We moved from zero-shot to few-shot after the first iteration, including samples from the training data excluded from the subset used to calculate evaluation metrics.

For the integrative evaluation stage, we performed a qualitative assessment of interrater discrepancies from the manual coding stage and misclassifications from the automation. We deliberated and explored different explanations. For instance, we examined the misclassifications in relation to the codebook, identifying the ways the concept is represented in the final operationalization (construct validity). We also returned to the manually coded data to check if the misclassifications related to errors in the original manual coding. Finally, we sought to assess whether the cause of identified construct

validity problems could be explained by problems with the concept of misunderstandings (concept validity). We only ended the integrative evaluation inference loop when both authors were satisfied with the validity assessment.

Applying the Noniterative Method

The manual coding stage for the noniterative method involved two coders scoring the full data set using the draft version of the codebook, created before conducting inference loops (Supplementary Materials B in the online supplemental materials). This codebook used our initial definition of misunderstanding with prototypical examples for both direct and felt misunderstanding. Of these, 2,000 sentences were shared among coders and used to calculate interrater reliability. Coders were provided with a single introductory meeting to discuss the codebook, and discrepancies were resolved by Alex Goddard.

For the classifier development stage, we employed the versions of the protocol implemented in the first inference loop conducted in the RAMP method (i.e., prior to any adjustments): for the rule-based classifier, we used the initial set of terms; for the supervised classifier, we used the initial hyperparameters (learning rate $= 5e-5$, batch size $= 32$, epochs $= 5$, proportion of data used for validation $= 30\%$); and for the LLM classifier, we used the initial zero-shot prompt. We withheld 30% from the development stage to evaluate the classifiers' performance on unseen data.

Transparency and Openness

All data have been made publicly available on the Open Science Framework (Goddard & Gillespie, 2025a) and can be accessed at https://osf.io/pe4jy/. The code behind the analysis has been made publicly available on GitHub (Goddard & Gillespie, 2025b) and can be accessed at https://github.com/alexiamhe93/RAMP_method via a Python Jupyter notebook. The GitHub repository also contains the final codebooks used for manual coding (Supplementary Materials A and B in the online supplemental materials).

Results

Manual Coding Stage

Table 4 reports the absolute agreement and interrater reliability of the coders during the throughput phase. In Loop 1, we found high agreement (95%) with weak reliability ($\alpha = .57$). This discrepancy was explained by the low proportion of misunderstandings (around 8%), because the absolute agreement can be very high by not coding for misunderstanding. Coders reported difficulties identifying misunderstandings due to a reported lack of context. At this stage, they were given randomly sampled sentences (i.e., decontextualized from the dialogue; see Supplementary Materials B in the online supplemental materials) and instructed to make a scoring decision solely on their lexical and semantic content. This was decided to be a cause of the low reliability, and accordingly, from Loop 2, the sentences were scored sequentially in the context of the dialogue.

Table 4
Manual Coding Throughput Interrater Reliability Results

Inference loop	Sample size (sentences)	Absolute agreement	Krippendorff's α	Gwet's AC1
1	713	0.95	.57	0.94
2	1,228	0.97	.71	0.97
3	1,101	0.97	.72	0.96
4	808	0.94	.78	0.92
5	862	0.98	.75	0.97

Note. AC1 = agreement coefficient 1.

Loop 2 saw a significant increase in the coder's reliability ($\alpha = .71$). The changes between Loop 2 and Loop 3 were more conservative, leading to a negligible increase in reliability (increase of .01). The changes to the codebook between these loops were nominal, focusing on the complexity of the operationalization. For Loop 4, the changes were more substantial and included the addition of a "Frequently Asked Questions" to provide clarification on edge cases of misunderstandings, such as what to do when uncertain, and how to address peculiarities of the scored texts (e.g., hashtags with Twitter data). These changes drove a larger increase in the reliability for Loop 4 ($\alpha = .78$). Loop 5 saw a drop in reliability following minor adjustments to the codebook. We thus decided to end the inference loop as discussions with coders did not reveal any new adjustments, and a further increase in reliability seemed unlikely.

The final reliability calculated in the output phase was higher than the previous loops on the training data ($\alpha = .79$, Gwet's AC1 $= 0.98$). Our final coding found 1,715 misunderstandings, approximately 8% of the sentences. Wikipedia Talk Pages contained the most misunderstandings ($n = 1{,}066$, 62%), with Reddit ($n = 373$, 22%) and Twitter Customer Support ($n = 276$, 16%) containing far fewer.

Classifier Development Stage

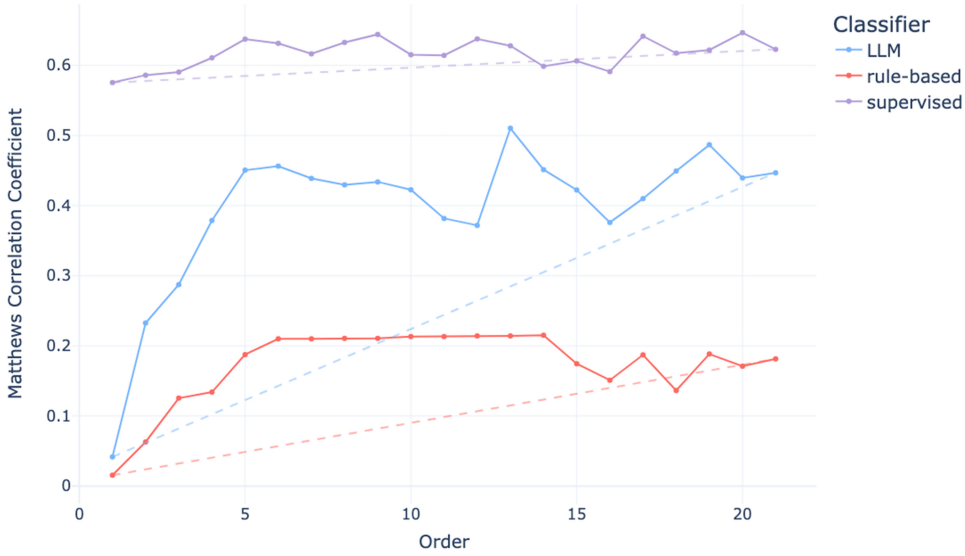
Figure 3 reports the MCC for the 21 iterations of the inference loop for each classifier. The statistics were calculated on validation data of different proportions, sampled from the training data. The MCC statistic was chosen because it is the most appropriate for highly skewed data (Chicco & Jurman, 2020). For the rule-based classifier, the performance was very low (MCC $= 0.02$) in the first loop, ramping up until Loop 5 but remaining low (MCC $= 0.19$). For Loops 6–15, we switched from counting words to word sequences. There was little change in performance after Loop 6 (MCC $= 0.21$), despite adding more patterns to boost recall. Eventually, we returned to counting words in Loop 16 in the goal of improving the performance, adjusting the items until the final loop (MCC $= 0.18$).

The supervised classifier started with moderate performance (MCC $= 0.58$) and ramped up marginally across all the iterations (maximum MCC $= 0.65$ for Loop 20). We found that lower learning rates, larger batch sizes, and larger number of epochs improved performance. Despite these marginal increases, the supervised classifier appeared to perform better than the rule-based or LLM classifiers. Finally, the LLM classifier followed a similar pattern to the rule-based classifier, where the performance started very low (MCC $= 0.05$) but ramped up rapidly until Loop 5 (MCC $= 0.45$) after we added examples and adjusted the prompt to the misclassifications. Unlike the rule-based classifier, the performance at this point had improved substantially. The subsequent iterations of the prompt yielded limited improvement until Loop 13 (MCC $= 0.51$); however, the prompt could not be improved upon in future iterations. The final loop ended with low-moderate performance (MCC $= 0.45$).

For the output phase, we used the withheld test data to compare the classifiers with the highest evaluation metrics from the throughput

Figure 3

Increases in Matthews Correlation Coefficient for the LLM, Rule-Based, and Supervised Classifier



Note. LLM = large language model. See the online article for the color version of this figure.

phase. For the rule-based classifier, we used the terms (word sequences) from Loop 14. For the supervised classifier, we trained a model using the training (70%) and validation data (30%) and used the hyperparameters of Loop 20 (learning rate = 2e − 5, batch size = 128, epochs = 4). For the LLM classifier, we used the prompt from Loop 13 (few-shot). Table 5 reports the output phase evaluation metrics for the three classifiers on the test data.

As expected, the supervised classifier performed best across every metric (MCC = 0.65). The supervised classifier had weaker precision (0.65) than recall (0.79), indicating that the model was identifying most misunderstandings but still generating many false positives. This pattern was the same for all three classifiers. The LLM classifier was second best and performed better than the rule-based classifier across most metrics. The rule-based classifier had poor precision and, therefore, a low MCC for misunderstanding. The weighted F1 score for the model was still high (0.89) despite the classifier's poor performance, demonstrating the importance of using appropriate evaluation metrics to the data balance.

Table 5

Performance of the RAMP Rule-Based, Supervised, and LLM Classifier on the Test Data (n = 6,420)

Measure	Rule-based	Supervised	LLM
Accuracy	0.89	0.95	0.89
Weighted F1 score (all)	0.89	0.95	0.90
Balanced accuracy	0.61	0.88	0.80
Precision (misunderstanding)	0.29	0.65	0.39
Recall (misunderstanding)	0.27	0.79	0.69
F1 score (misunderstanding)	0.28	0.71	0.50
MCC	0.22	0.69	0.47

Note. RAMP = Repeated Adjustment of Measurement Protocols; LLM = large language model; F1 = harmonic mean of precision and recall; MCC = Matthews correlation coefficient.

Integrative Evaluation Stage

There were two surprising findings from the previous phases. First, the LLM classifier performed worse than the supervised classifier despite being easier to adjust (Figure 3). Second, we identified sentences in both the interrater discrepancies and false positive misclassifications that involved overt corrections of other's assumed misunderstanding (Table 6). These were surprising because the codebook dissuades coders from scoring corrections as they are not misunderstandings. Instead, they evidence that the self has understood the other's misunderstanding in a previous turn. These two surprising findings appear related as the LLM identified corrections more frequently than the supervised classifier. Our analysis of the false negatives, however, did not reveal any additional surprises.

To explain the surprising findings, we determined that misunderstandings have a concept validity problem. Specifically, they were explainable by a conceptual shift away from misunderstandings toward conversational repairs (Schegloff et al., 1977). Conversational repairs conceptualize the systematic ways people address miscommunications and misunderstandings in dialogue. The interrater discrepancies and misclassifications in Table 6 were better interpreted as instances of other repair, where a self informs the other that they have misunderstood something (Collister, 2011).

Other repairs do not neatly fit into our operationalization as they are neither direct nor felt misunderstanding and, instead, are a correction of someone else's misunderstanding. However, direct and felt misunderstandings can be integrated into the concept of repair. A direct misunderstanding is an other-initiated repair, where the other seeks clarification after misunderstanding the self (e.g., "What do you mean?"). A felt misunderstanding is a third turn self repair, where the self notes the other has misunderstood their previous statement and corrects them (e.g., "I didn't mean to say that"). The repair typology provides a broader set of constructs that incorporate, but go beyond, direct and felt misunderstandings.

Table 6
Surprising Interrater Discrepancies in the Manual Coding and Supervised Classifier Misclassifications

Type	Text
Interrater discrepancies	"You seem to have misread the guidelines," "I mean yeah but your only thinking of it as an asset and not as a character," "This is AMC, but I feel you," "not by that title"
Supervised false positives	"It didn't come from Lake Robert," "I still feel that you shouldn't have sub-paged without consensus," "No, he didn't."
LLM false positives	"Your arm is only from your elbow to your shoulder," "You don't have layers of dab pages," "Don't fool yourself, that's what's going on," "See, totally not kids in cages," "You are 'not' the owner of the article," "Some of your comments are not helping," "Not quite," "They weren't 'trying to be unique', they were broken in eyes of majority"

Note. LLM = large language model.

The concept of misunderstanding is not invalid per se. Instead, it is that misunderstandings are an internal state which repairs make visible (Schegloff, 1992). Counterintuitively, this means repairs are a better concept for operationalizing misunderstanding in text. The problem is that once a participant explicitly indicates there has been a misunderstanding, they are in the process of repairing it. For instance, direct misunderstandings are asking for clarification to the self's misunderstanding and felt misunderstandings are clarifying the other's misunderstanding. Furthermore, people might misunderstand each other and not say anything. These misunderstandings are undetectable to an observer and, therefore, cannot be quantified directly. Measuring repairs side-steps this problem and focuses only on misunderstandings that have been made explicit by the participants themselves through seeking understanding.

This conceptual problem provided a reasonable explanation for why the supervised classifier performed better than the LLM classifier. Repairs are simply more salient in the text than repairs. This means that the coded sentences may have been semantically similar enough for a supervised classifier to learn, but not conceptually coherent enough to be correctly recognized by the LLM classifier's prompt. Describing how to identify misunderstandings in a prompt could have introduced conceptual confusion that is absent for the supervised classifier.

A consequence of this concept validity problem was that it called into question the construct validity of our supervised classifier. However, the classifier was generally able to differentiate misunderstanding sentences from nonmisunderstanding sentences, despite the underlying conceptual problems. It provided a proof of concept that supervised classification using deep learning is a promising avenue for measuring repairs. It also feeds forward into future research, which should consider scoring repairs at a turn level, reflecting directions from the literature (e.g., Dingemanse et al., 2016). But, the important point for our current purposes is how, during the integrative evaluation stage, examination of challenges in the measurement process can backpropagate insights to the initiating concepts.

Comparing RAMP and Noniterative Methods

For the manual coding stage, the noniterative coders scored 592 (3%) sentences for misunderstandings, far fewer than the RAMP coders. The noniterative coders had lower reliability ($\alpha = .28$, Gwet's AC1 = 0.95) and agreement (95%) than those reported in the RAMP output phase. Instead, they resembled Loop 1 of RAMP's throughput phase, which is unsurprising given that the same codebook was employed. The noniterative coders disagreed with the RAMP coders on 1,453 (7%) sentences. Of these, the majority were cases where noniterative coders did not score for misunderstanding, but RAMP coders did ($n = 1{,}288$, 88.6%). Most of the misunderstandings (452, 72%) identified by noniterative coders were also identified by the RAMP coders. This indicates that the iterative RAMP method led to more inclusive operationalizations and coders becoming better able to identify more subtle forms of misunderstanding.

For the classifier development stage, Table 7 reports the evaluation metrics for the three noniterative classifiers on the test data, withheld from the supervised classifier's training. All three classifiers performed worse than the RAMP classifiers across most metrics (Table 5). For the rule-based and supervised classifier, the accuracy and weighted F1 scores were marginally higher; however, this is explained by the statistics sensitivity to imbalanced data and the lower frequency of misunderstandings in the noniterative data. Like the RAMP classifiers, the supervised classifier performed best. Contrasting with the RAMP classifiers, the LLM classifier was the worst performing, with the negative MCC (−0.02) indicating that the model's predictions were worse than chance.

Case Study Limitations

Our case study had limitations during the manual coding and classifier development stages. Regarding the manual coding, additional independent coders (external to training) could have been employed to test the generalizability of the final codebook. However, the

Table 7
Performance of the Noniterative Rule-Based, Supervised, and LLM Classifier on the Test Data ($n = 6{,}595$)

Measure	Rule-based	Supervised	LLM
Accuracy	**0.97**	**0.97**	0.65
Weighted F1 score (all)	**0.96**	**0.97**	0.77
Balanced accuracy	0.51	0.64	0.47
Precision (misunderstanding)	0.07	0.46	0.02
Recall (misunderstanding)	0.02	0.30	0.28
F1 score (misunderstanding)	0.03	0.36	0.04
MCC	0.03	0.35	−0.02

Note. Metrics where noniterative classifiers performed better than RAMP classifiers are in bold. LLM = large language model; F1 = harmonic mean of precision and recall; MCC = Matthews correlation coefficient; RAMP = Repeated Adjustment of Measurement Protocols.

operationalizations were complex, and the new coders would likely have required training, potentially driving further changes to the codebook and invalidating the point of using an independent group. In addition, some sentences in the final interrater reliability subsample had previously been seen by the coders in previous training loops. During the manual coding stage, keeping the calibration data fully separate from the full data would have provided an extra layer of rigor comparable to the train-test splitting practices in data science (and in the RAMP classifier development stage). Finally, misunderstandings were operationalized as a binary variable, rather than categorical. We did this to demonstrate the RAMP method on a minimal case of binary classification; however, future studies should explore operationalizing misunderstandings at both direct and felt levels (categorical).

Regarding the classifier development, the LLM classifier's model (GPT-4o) is proprietary, meaning its training process and data are not open to academic scrutiny, resulting in potentially hidden biases (Liesenfeld et al., 2023). We opted for GPT-4o because, at the time of writing, it was the most accurate model across multiple tasks (OpenAI et al., 2024). Future studies should explore the use of open-source LLMs that are becoming increasingly capable. Another limitation is that we did not explore the role of temperature—the parameter that determines variability—in developing the LLM classifier. If classifications remain reliable across inference loops with higher temperature levels, confidence is gained on the prompt's ability to produce correct classifications. Additionally, different training examples included in the prompt can produce varied results, adding an additional parameter to explore during LLM classifier development. Finally, our limited number of loops in the classifier development stage provided an arbitrary ceiling for development.

Discussion

We have introduced the RAMP method for developing text classifiers for psychological text analysis. The method proposes using inference loops to integrate best practices of content analysis (Krippendorff, 2019), data science (Donoho, 2017), and psychometric validation using abduction (Gillespie et al., 2024). The inference loop offers a flexible framework for integrating the three stages and highlights how repeated adjustments following new evidence relate to validity in scientific processes and findings. All three stages of RAMP rely on inference loops that enable an iterative assessment of concept and construct validity. We showed how the inference loop operates within three different approaches to text classification (rule-based, supervised machine learning, and LLM) and demonstrated its efficacy compared to a noniterative method. RAMP formalizes how validity is an ongoing dynamic process of adjustment both within and across studies.

Manual coding, classifier development, and integrative evaluation stages are rarely performed together in text classification (Song et al., 2020). Our case study highlights why RAMP's integration of the three stages is advantageous for establishing concept and construct validity. Without understanding how misunderstandings were operationalized in the manual coding, identifying the concept validity problems relating to conversational repairs would have been difficult. We could have been fooled at the manual coding and classifier development stage by the statistical outputs into thinking our best classifier was valid. In manual coding, Gwet's AC1 was very high and Krippendorff's α reasonably high considering the imbalance of the categories. This created misplaced security as to the construct validity of the operationalization, revealing why interrater reliability is not a direct measure of validity, and only a minimal requirement (Krippendorff, 2019). A similar statement also applies to evaluation metrics; in the classifier development stage, the accuracy and weighted F1 scores were deceptively high (a result of data imbalance), requiring us to use the more robust MCC statistic to evaluate the classifier's performance.

Without proper qualitative scrutiny of the results during RAMP's integrative evaluation stage, there could have been a significant risk of misinterpreting statistics for construct validity. Our codebook and classifier are not measuring misunderstandings, but rather attempts to maintain mutual understanding, the conceptual opposite of misunderstanding. Because of this concept validity problem, the supervised classifier does not have construct validity, regardless of its evaluation metrics. This challenge arises because the measured concept (misunderstandings) is a psychological state that sits behind the text. Once verbalized—and therefore measurable—the original concept changes form (resolving of misunderstanding) and is better expressed as a different concept (repairs). This validity challenge needs to be kept in mind when trying to measure psychological phenomena in observable manifestations (e.g., text).

The RAMP method, by introducing inference loops, goes beyond a simple integration of stages. The inference loop applies within and across different iterations of RAMP, emphasizing a dynamic definition of validity as an ongoing process, rather than a static state. Validating empirical findings in psychology requires intersubjective agreement on concepts (Muthukrishna & Henrich, 2019). Currently, there is not enough scrutiny of concepts or constructs in psychology (Bringmann et al., 2022; Flake & Fried, 2020), and questions of validity are often ignored, even in replication studies (Flake et al., 2022).

By powering RAMP with inference loops, we seek to encourage a common task framework in the development of textual constructs, similar to the one found in data science for developing algorithms (Donoho, 2017). For instance, an initial RAMP iteration (with associated data) could be made public with a competition to perform a further iteration that addresses the identified validity issues. Competitors would perform their own manual coding, classifier development, and integrative evaluation, with an independent panel of judges determining the most compelling case for a valid construct. Each competition would seek to maximize evaluation metrics (construct validity) and justify any operational and theoretical changes from previous iterations (concept validity). This process could be repeated until there is a widespread intersubjective agreement on the concept and construct validity of a measure. Such a collaborative process would contribute toward a more robust psychological science. To support the use of RAMP, we have included a checklist for its reporting (Appendix).

RAMP Limitations

There are three key limitations and avenues for future research relating to the RAMP method. First, it makes no use of external validity checks, such as convergent or discriminant validity, which enable more robust evaluation than internal validity checks alone (Birkenmaier et al., 2024). For instance, we could have employed another measure of mutual understanding (e.g., linguistic alignment; Duran et al., 2019) to assess the convergent validity of our best misunderstandings classifier. However, this was deliberate because RAMP and the inference loops were designed to maximize concept

and construct validity through conventional practices in classifier development. RAMP is not a definitive list of statistical practices for evaluating classifiers, but rather an overarching framework for their implementation. Nonetheless, future iterations on RAMP could be expanded to include external validity checks.

The second limitation is that RAMP was only demonstrated on a binary classification case, with limited discussion of categorical, ordinal, or continuous classification and unsupervised methods. RAMP's flexible design around inference loops means different reliability and accuracy measures can easily be integrated into the framework. The case of unsupervised classification could also utilize the RAMP framework. For example, when developing topic models, researchers are recommended to select models using both statistical information and qualitative assessments based on the interpretability of the algorithms clusters (Laureate et al., 2023). This process also entails a repeated adjustment and, therefore, suggests potential for integrating RAMP with unsupervised classification.

Third, RAMP was only validated by comparing it to a noniterative method using the same data, and further evaluations are required to verify the method's utility. Future studies should employ RAMP for different research contexts, comparing its performance on different types of text data (e.g., diaries, emails, and open-text questions), different types of variables (e.g., latent and observable), and applications (e.g., clinical and organizational). Further comparisons with alternative methods are also required. For instance, a RAMP measure could be assessed against a measure developed using three different researchers to coordinate the manual coding, classification, and integrative evaluation stages independently. A third party could then compare the quantitative results and their qualitative interpretation.

Conclusion

This article introduced the RAMP method for developing and validating psychological text classifiers. It combines manual coding, classifier development, and integrative evaluation stages into a single framework designed for establishing construct and concept validity. Each stage is powered by an inference loop, where researchers repeatedly adjust their approach given new evidence. The inference loop not only allows for the integration of the different stages of RAMP but also provides a flexible structure for evaluating validity both within and across studies. Integrating the best practices of content analysis and data science together with psychometric validation using abduction emphasizes that measure development is a process of repeatedly adjusting concepts and constructs, rather than a single-shot evaluation of independent parts. There is always the possibility that the manual coding was dubious, regardless of interrater reliability, and that classifiers are reproducing conceptual confusion, regardless of their evaluation metrics. An integrated framework for evaluating validity across the classifier development pipeline provides an opportunity for detecting and avoiding these issues early and thus improving the validity of text classifiers.

References

Albert, S., & de Ruiter, J. P. (2018). Improving human interaction research through ecological grounding. *Collabra: Psychology*, *4*(1), 1–24. https://doi.org/10.1525/collabra.132

Andersen, J. P. (2025). Applied research is the path to legitimacy in psychological science. *Nature Reviews Psychology*, *4*(1), 1–2. https://doi.org/10.1038/s44159-024-00388-9

Anderson, C. (2008, June 23). The end of theory: The data deluge makes the scientific method obsolete. *Wired*. https://www.wired.com/2008/06/pb-theory/

Thought Vector, & Axelbrooke, S. (2017). *Customer support on Twitter (v10)*. https://kaggle.com/thoughtvector/customer-support-on-twitter

Bahrami, M., Mansoorizadeh, M., & Khotanlou, H. (2023). *Few-shot learning with prompting methods*. 2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA) (pp. 1–5). https://doi.org/10.1109/IPRIA59240.2023.10147172

Banks, G. C., O'Boyle, E. H., Pollack, J. M., White, C. D., Batchelor, J. H., Whelpley, C. E., Abston, K. A., Bennett, A. A., & Adkins, C. L. (2016). Questions about questionable research practices in the field of management: A guest commentary. *Journal of Management*, *42*(1), 5–20. https://doi.org/10.1177/0149206315619011

Beauchamp, N. (2020). Modeling and measuring deliberation online. In B. Foucault Welles & S. González-Bailón (Eds.), *The Oxford handbook of networked communication* (pp. 320–349). Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190460518.013.23

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 (pp. 610–623). https://doi.org/10.1145/3442188.3445922

Biggiogera, J., Boateng, G., Hilpert, P., Vowels, M., Bodenmann, G., Neysari, M., Nussbeck, F., & Kowatsch, T. (2021). *BERT meets LIWC: Exploring state-of-the-art language models for predicting communication behavior in couples' conflict interactions*. Companion Publication of the 2021 International Conference on Multimodal Interaction (pp. 385–389). https://doi.org/10.1145/3461615.3485423

Birkenmaier, L., Lechner, C. M., & Wagner, C. (2024). The search for solid ground in text as data: A systematic review of validation practices and practical recommendations for validation. *Communication Methods and Measures*, *18*(3), 249–277. https://doi.org/10.1080/19312458.2023.2285765

Boyd, R. L., Ashokkumar, A., Seraj, S., & Pennebaker, J. (2022). *The development and psychometric properties of LIWC-22*. The University of Texas at Austin. https://doi.org/10.13140/RG.2.2.23890.43205

Boyd, R. L., & Schwartz, H. A. (2021). Natural language analysis and the psychology of verbal behavior: The past, present, and future states of the field. *Journal of Language and Social Psychology*, *40*(1), 21–41. https://doi.org/10.1177/0261927X20967028

Bringmann, L. F., Elmer, T., & Eronen, M. I. (2022). Back to basics: The importance of conceptual clarification in psychological science. *Current Directions in Psychological Science*, *31*(4), 340–346. https://doi.org/10.1177/09637214221096485

Chen, J., Qiu, L., & Ho, M.-H. R. (2020). A meta-analysis of linguistic markers of extraversion: Positive emotion and social process words. *Journal of Research in Personality*, *89*(10), Article 104035. https://doi.org/10.1016/j.jrp.2020.104035

Chernyavskiy, A., Ilvovsky, D., & Nakov, P. (2021). Transformers: "The end of history" for natural language processing? In N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, & J. A. Lozano (Eds.), *Machine learning and knowledge discovery in databases. Research track* (pp. 677–693). Springer International Publishing. https://doi.org/10.1007/978-3-030-86523-8_41

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), Article 6. https://doi.org/10.1186/s12864-019-6413-7

Collister, L. B. (2011). *-repair in online discourse. *Journal of Pragmatics*, *43*(3), 918–921. https://doi.org/10.1016/j.pragma.2010.09.025

Corti, K., & Gillespie, A. (2016). Co-constructing intersubjectivity with artificial conversational agents: People are more likely to initiate repairs of misunderstandings with agents represented as human. *Computers in Human Behavior*, *58*, 431–442. https://doi.org/10.1016/j.chb.2015.12.039

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin*, *52*(4), 281–302. https://doi.org/10.1037/h0040957

Danescu-Niculescu-Mizil, C., Lee, L., Pang, B., & Kleinberg, J. (2012). *Echoes of power: Language effects and power differences in social interaction*. Proceedings of the 21st International Conference on World Wide Web (pp. 699–708). https://doi.org/10.1145/2187836.2187931

Demszky, D., Yang, D., Yeager, D. S., Bryan, C. J., Clapper, M., Chandhok, S., Eichstaedt, J. C., Hecht, C., Jamieson, J., Johnson, M., Jones, M., Krettek-Cobb, D., Lai, L., JonesMitchell, N., Ong, D. C., Dweck, C. S., Gross, J. J., & Pennebaker, J. W. (2023). Using large language models in psychology. *Nature Reviews Psychology*, *2*, 688–701. https://doi.org/10.1038/s44159-023-00241-5

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423

Dingemanse, M., Kendrick, K. H., & Enfield, N. J. (2016). A coding scheme for other-initiated repair across languages. *Open Linguistics*, *2*(1), 35–46. https://doi.org/10.1515/opli-2016-0002

Donoho, D. (2017). 50 Years of data science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766. https://doi.org/10.1080/10618600.2017.1384734

Duran, N. D., Paxton, A., & Fusaroli, R. (2019). ALIGN: Analyzing Linguistic Interactions with Generalizable techNiques—A python library. *Psychological Methods*, *24*(4), 419–438. https://doi.org/10.1037/met0000206

Eichstaedt, J. C., Kern, M. L., Yaden, D. B., Schwartz, H. A., Giorgi, S., Park, G., Hagan, C. A., Tobolsky, V. A., Smith, L. K., Buffone, A., Iwry, J., Seligman, M. E. P., & Ungar, L. H. (2021). Closed- and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. *Psychological Methods*, *26*(4), 398–427. https://doi.org/10.1037/met0000349

Eronen, M. I., & Bringmann, L. F. (2021). The theory crisis in psychology: How to move forward. *Perspectives on Psychological Science*, *16*(4), 779–788. https://doi.org/10.1177/1745691620970586

Feng, G. C. (2014). Intercoder reliability indices: Disuse, misuse, and abuse. *Quality & Quantity*, *48*(3), 1803–1815. https://doi.org/10.1007/s11135-013-9956-8

Feng, Y., & Hancock, G. R. (2022). Model-based incremental validity. *Psychological Methods*, *27*(6), 1039–1060. https://doi.org/10.1037/met0000342

Flake, J. K., Davidson, I. J., Wong, O., & Pek, J. (2022). Construct validity and the validity of replication studies: A systematic review. *American Psychologist*, *77*(4), 576–588. https://doi.org/10.1037/amp0001006

Flake, J. K., & Fried, E. I. (2020). Measurement Schmeasurement: Questionable measurement practices and how to avoid them. *Advances in Methods and Practices in Psychological Science*, *3*(4), 456–465. https://doi.org/10.1177/2515245920952393

Freeman, E. (1973). Objectivity as "intersubjective agreement." *The Monist*, *57*(2), 168–175. https://doi.org/10.5840/monist19735722

Friess, D., & Eilders, C. (2015). A systematic review of online deliberation research. *Policy & Internet*, *7*(3), 319–339. https://doi.org/10.1002/poi3.95

Gillespie, A., Glăveanu, V., & de Saint-Laurent, C. (2024). *Pragmatism and methodology: Doing research that matters with mixed methods*. Cambridge University Press.

Gisev, N., Bell, J. S., & Chen, T. F. (2013). Interrater agreement and interrater reliability: Key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, *9*(3), 330–338. https://doi.org/10.1016/j.sapharm.2012.04.004

Goddard, A., & Gillespie, A. (2023). Textual indicators of deliberative dialogue: A systematic review of methods for studying the quality of online dialogues. *Social Science Computer Review*, *41*(6), 2364–2385. https://doi.org/10.1177/08944393231156629

Goddard, A., & Gillespie, A. (2025a). *Data for The Repeated Adjustment of Measurement Protocols (RAMP) method for developing high-validity text classifiers*. https://doi.org/10.17605/OSF.IO/PE4JY

Goddard, A., & Gillespie, A. (2025b). *RAMP_V1.0.0* (Version accepted) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.16962563

González-Carvajal, S., & Garrido-Merchán, E. C. (2021). *Comparing BERT against traditional machine learning text classification* (No. arXiv:2005.13012). arXiv. https://doi.org/10.48550/arXiv.2005.13012

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org/

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. https://doi.org/10.1093/pan/mps028

Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing*, *28*(3), 539–547. https://doi.org/10.1007/s11222-017-9746-6

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, *1*(1), 77–89. https://doi.org/10.1080/19312450709336664

Heasman, B., & Gillespie, A. (2018). Perspective-taking is two-sided: Misunderstandings between people with Asperger's syndrome and their family members. *Autism*, *22*(6), 740–750. https://doi.org/10.1177/1362361317708287

Holtzman, N. S., Tackman, A. M., Carey, A. L., Brucks, M. S., Küfner, A. C. P., Deters, F. G., Back, M. D., Donnellan, M. B., Pennebaker, J. W., Sherman, R. A., & Mehl, M. R. (2019). Linguistic markers of grandiose narcissism: A LIWC analysis of 15 samples. *Journal of Language and Social Psychology*, *38*(5–6), 773–786. https://doi.org/10.1177/0261927X19871084

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2022). *Spacy: Industrial-strength natural language processing in Python*. Explosion.

Jackson, J. C., Watts, J., List, J.-M., Puryear, C., Drabble, R., & Lindquist, K. A. (2022). From text to thought: How analyzing language can advance psychological science. *Perspectives on Psychological Science*, *17*(3), 805–826. https://doi.org/10.1177/17456916211004899

Jun, H., Peng, L., Changhui, J., Pengzheng, L., Shenke, W., & Kejia, Z. (2021). *Personality classification based on BERT model*. 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT) (pp. 150–152). https://doi.org/10.1109/ICESIT53460.2021.9697048

Krippendorff, K. (1970). Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, *30*(1), 61–70. https://doi.org/10.1177/001316447003000105

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

Kumar, A., & Jain, A. K. (2022). Emotion detection in psychological texts by fine-tuning BERT using emotion–cause pair extraction. *International Journal of Speech Technology*, *25*(3), 727–743. https://doi.org/10.1007/s10772-022-09982-9

Laing, R. D., Phillipson, H., & Lee, A. R. (1966). *Interpersonal perception: A theory and a method of research* (p. vii, 179). Springer.

Laureate, C. D. P., Buntine, W., & Linger, H. (2023). A systematic review of the use of topic models for short text social media analysis. *Artificial Intelligence Review*, *56*(12), 14223–14255. https://doi.org/10.1007/s10462-023-10471-x

Lees, J., & Cikara, M. (2020). Inaccurate group meta-perceptions drive negative out-group attributions in competitive contexts. *Nature Human Behaviour*, *4*(3), 279–286. https://doi.org/10.1038/s41562-019-0766-4

Liesenfeld, A., Lopez, A., & Dingemanse, M. (2023). *Opening up ChatGPT: Tracking openness, transparency, and accountability in instruction-tuned text generators*. Proceedings of the 5th International Conference on Conversational User Interfaces (pp. 1–6). https://doi.org/10.1145/3571884.3604316

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*(6), 335–346. https://doi.org/10.1038/s41583-020-0277-3

Livingstone, A. G., Lucía, F. R., & Rothers, A. (2020). "They just don't understand us": The role of felt understanding in intergroup relations. *Journal of Personality and Social Psychology*, *119*(3), 633–656. https://doi.org/10.1037/pspi0000221

Locke, E. A. (2012). Construct validity versus concept validity. *Human Resource Management Review*, *22*(2), 146–148. https://doi.org/10.1016/j.hrmr.2011.11.008

MacPhail, C., Khoza, N., Abler, L., & Ranganathan, M. (2016). Process guidelines for establishing Intercoder Reliability in qualitative studies. *Qualitative Research*, *16*(2), 198–212. https://doi.org/10.1177/1468794115577012

Mitchell, T., Buchanan, B., DeJong, G., Dietterich, T., Rosenbloom, P., & Waibel, A. (1990). Machine learning. *Annual Review of Computer Science*, *4*(1), 417–433. https://doi.org/10.1146/annurev.cs.04.060190.002221

Muthukrishna, M., & Henrich, J. (2019). A problem in theory. *Nature Human Behaviour*, *3*(3), 221–229. https://doi.org/10.1038/s41562-018-0522-1

Nadeau, D., Kroutikov, M., McNeil, K., & Baribeau, S. (2024). *Benchmarking Llama2, Mistral, Gemma and GPT for factuality, toxicity, bias and propensity for hallucinations* (No. arXiv:2404.09785). arXiv. https://doi.org/10.48550/arXiv.2404.09785

Neuendorf, K. A. (2017). *The content analysis guidebook* (2nd ed.). Sage Publications.

Neuendorf, K. A. (2018). Content analysis and thematic analysis. In P. Brough (Ed.), *Advanced research methods for applied psychology: Design, analysis, and reporting* (pp. 211–223). Routledge.

Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(11), 2600–2606. https://doi.org/10.1073/pnas.1708274114

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), Article aac4716. https://doi.org/10.1126/science.aac4716

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024). *GPT-4 Technical Report* (No. arXiv:2303.08774). arXiv. https://doi.org/10.48550/arXiv.2303.08774

Park, N., & Peterson, C. (2006). Character strengths and happiness among young children: Content analysis of parental descriptions. *Journal of Happiness Studies*, *7*(3), 323–341. https://doi.org/10.1007/s10902-005-3648-6

Peirce, C. S. (1955). *Philosophical writings of Peirce*. Dover Publications.

Peirce, C. S. (1965). Pragmatism and abduction. In C. Hartshorne & P. Weiss (Eds.), *Collected papers of Charles Sanders Peirce* (Vol. 5, pp. 112–131). Belknap Press of Harvard University Press.

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*. University of Texas, Austin.

Pennebaker, J. W., & Francis, M. E. (1999). *Linguistic Inquiry and Word Count (LIWC): A computer-based text analysis program*. Lawrence Erlbaum.

Popper, K. R. (1959). *The logic of scientific discovery*. Hutchinson.

Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019). *An overview of bag of words; Importance, implementation, applications, and challenges*. 2019 International Engineering Conference (IEC) (pp. 200–204). https://doi.org/10.1109/IEC47844.2019.8950616

Ragsdale, J. M., Christiansen, N. D., Frost, C. T., & Rahael, J. A. (2013). Content analysis of personality at work. In N. D. Christiansen & R. Tett (Eds.), *Handbook of personality at work* (pp. 498–522). Routledge.

Rathje, S., Mirea, D.-M., Sucholutsky, I., Marjieh, R., Robertson, C., & Bavel, J. J. V. (2023). *GPT Is an effective tool for multilingual psychological text analysis*. OSF. https://doi.org/10.31234/osf.io/sekf5

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *'Why should I trust you?': Explaining the predictions of any classifier* (No. arXiv:1602.04938). arXiv. https://doi.org/10.48550/arXiv.1602.04938

Rosenbusch, H., Soldner, F., Evans, A. M., & Zeelenberg, M. (2021). Supervised machine learning methods in psychology: A practical introduction with annotated R code. *Social and Personality Psychology Compass*, *15*(2), 1–25. https://doi.org/10.1111/spc3.12579

Rubin, R. B. (1994). Feelings of understanding/misunderstanding scale. In R. B. Rubin, P. Palmgreen, & H. E. Sypher (Eds.), *Communication research measures: A sourcebook* (pp. 165–168). Guilford Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Schegloff, E. A. (1992). Repair after next turn: The last structurally provided defense of intersubjectivity in conversation. *American Journal of Sociology*, *97*(5), 1295–1345. https://doi.org/10.1086/229903

Schegloff, E. A., Jefferson, G., & Sacks, H. (1977). The preference for self-correction in the organization of repair in conversation. *Language*, *53*(2), 361–382. https://doi.org/10.2307/413107

Shiraishi, N., & Reilly, J. (2022). Content analysis of the emotions affecting caregivers of relatives with schizophrenia. *Current Psychology*, *41*(10), 6755–6765. https://doi.org/10.1007/s12144-020-01185-2

Silva, E. S., & Hassani, H. (2023). The role of ChatGPT in data science: How AI-assisted conversational interfaces are revolutionizing the field. *Big Data and Cognitive Computing*, *7*(2), Article 62. https://doi.org/10.3390/bdcc7020062

Smith, L. N. (2018). *A disciplined approach to neural network hyper-parameters: Part 1—Learning rate, batch size, momentum, and weight decay* (No. arXiv:1803.09820). arXiv. https://doi.org/10.48550/arXiv.1803.09820

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. https://doi.org/10.1080/10584609.2020.1723752

Stone, P. J., Dunphy, D. C., & Smith, M. S. (1966). *The General Inquirer: A computer approach to content analysis* (p. 651). MIT Press.

Stromer-Galley, J. (2007). Measuring deliberation's content: A coding scheme. *Journal of Deliberative Democracy*, *3*(1), 1–35. https://doi.org/10.16997/jdd.50

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? In M. Sun, X. Huang, H. Ji, Z. Liu, & Y. Liu (Eds.), *Chinese computational linguistics* (pp. 194–206). Springer International Publishing. https://doi.org/10.1007/978-3-030-32381-3_16

Tackman, A. M., Sbarra, D. A., Carey, A. L., Donnellan, M. B., Horn, A. B., Holtzman, N. S., Edwards, T. S., Pennebaker, J. W., & Mehl, M. R. (2019). Depression, negative emotionality, and self-referential language: A multi-lab, multi-measure, and multi-language-task research synthesis. *Journal of Personality and Social Psychology*, *116*(5), 817–834. https://doi.org/10.1037/pspp0000187

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54. https://doi.org/10.1177/0261927X09351676

Timmermans, S., & Tavory, I. (2022). *Data analysis in qualitative research: Theorizing with abductive analysis*. University of Chicago Press. https://press.uchicago.edu/ucp/books/book/chicago/D/bo133273407.html

Urban, C. J., & Gates, K. M. (2021). Deep learning: A primer for psychologists. *Psychological Methods*, *26*(6), 743–773. https://doi.org/10.1037/met0000374

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. https://doi.org/10.1080/19312458.2020.1869198

van Dijk, T. A. (1977). *Text and context: Explorations in the semantics and pragmatics of discourse*. Longman Group UK.

Webb, E. J., Campbell, D. T., Schwartz, R. D., & Sechrest, L. (1966). *Unobtrusive measures: Nonreactive research in the social sciences* (p. xii, 225). Rand Mcnally.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Rush, A. M. (2020). *Huggingface's transformers: State-of-the-art natural language processing* (No. arXiv:1910.03771). arXiv. https://doi.org/10.48550/arXiv.1910.03771

Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2023). *Can large language models transform computational social science?* (No. arXiv:2305.03514). arXiv. https://doi.org/10.48550/arXiv.2305.03514

Appendix

Repeated Adjustment of Measurement Protocols Checklist

Item	Checklist item	Item description
Manual coding stage		
1	Define the concepts	Provide an explicit definition of the concepts and their theoretical background(s).
1	Justification for measurement	Provide details of how concepts have been operationalized in previous studies, explain why a new measurement protocol for the concepts is needed.
2	Raw text data	Describe the type of documents used (e.g., sentences and paragraphs), how they were collected, how they relate to the target concepts, and any cleaning and/or anonymization procedures applied prior to analysis.
3	Initial codebook	Summarize the initial codebook.
4	Interrater reliability method(s)	Provide details of how interrater reliability was quantified, the number of coders, justification for using these specific coders, and the proportion of data withheld for validation.
5	Coder training	Describe how coders were trained, including their role in troubleshooting interrater discrepancies and adjusting the codebook.
6	Interrater reliability (calibration data)	Report the changes in interrater reliability during training.
7	Codebook adjustments and finalization	Describe how and why the codebook was adjusted across the training iterations and why the training was ended.
8	Final codebook	Summarize final operationalization and describe where the final codebook can be accessed.
9	Interrater reliability (full data)	Report the interrater reliability on the withheld test data and describe how coders scored the full data set.
10	Interrater discrepancies	Describe and provide examples of interrater discrepancies.
11	Coded data set	Describe the distribution of the concepts in the final coded data set.
Classifier development stage		
12	Coded data set*	Describe the type of documents used (e.g., sentences and paragraphs), how they were coded for the target concept, details of any interrater reliability calculations, and justification for using it.
13	Initial protocol	Describe and justify the type of classifiers chosen and the quantitative (e.g., hyperparameters and choice of model) and qualitative (e.g., changes in prompts or dictionary items) parameters adjusted during training.
14	Statistical evaluation methods	Describe and justify the how performance was quantified using evaluation metrics.
15	Statistical evaluation (validation data)	Report the changes in evaluation metrics during training.
16	Protocol adjustments and finalization	Describe how and why the protocol was adjusted across the development iterations and why the development was ended.
17	Final protocol	Summarize final protocol, provide details of implementation, provide instructions for replication.
18	Statistical evaluation (test data)	Report the evaluation metrics on the withheld test data.
19	Misclassifications*	Describe and provide examples of misclassifications.
Integrative evaluation stage		
20	Interrater discrepancies*	Describe and provide examples of interrater discrepancies.
21	Misclassifications*	Describe and provide examples of misclassifications.
23	Surprising observations	Describe what surprised you about the process of operationalizing a concept.
24	Explaining surprises	Discuss potential adjustments to the concept and/or operationalizations which would explain the surprises (e.g., use a different concept).
25	Construct validity summary	Summarize the evidence of construct validity (does the protocol measure what it's intended to measure?).
26	Concept validity summary	Summarize the evidence for concept validity (does my definition of the concept have intersubjective agreement between scientists?)

Note. The asterisk "*" provide details if not reported on in previous stage or make use of information provided from previous stage.

Received May 31, 2024
Revision received June 16, 2025
Accepted July 7, 2025 ■