Andersen, Nico; Zehner, Fabian; Goldhammer, Frank

Semi-automatic coding of open-ended text responses in large-scale assessments

Journal of computer assisted learning 39 (2022) 3, S. 841-854

Received: 2 February 2022 | Revised: 6 May 2022 | Accepted: 14 June 2022

DOI: 10.1111/jcal.12717

REVIEW ARTICLE

Journal of Computer Assisted Learning WILEY

Semi-automatic coding of open-ended text responses in large-scale assessments

Nico Andersen[1] | Fabian Zehner[1,2] | Frank Goldhammer[1,2]

[1]DIPF | Leibniz Institute for Research and Information in Education, Frankfurt am Main, Germany

[2]Centre for International Student Assessment (ZIB) e.V., Frankfurt am Main, Germany

Correspondence
Nico Andersen, DIPF | Leibniz Institute for Research and Information in Education, Rostocker Str. 6, 60323 Frankfurt am Main, Germany.
Email: andersen.nico@dipf.de

Abstract

Background: In the context of large-scale educational assessments, the effort required to code open-ended text responses is considerably more expensive and time-consuming than the evaluation of multiple-choice responses because it requires trained personnel and long manual coding sessions.

Aim: Our semi-supervised coding method *eco* (exploring coding assistant) dynamically supports human raters by automatically coding a subset of the responses.

Method: We map normalized response texts into a semantic space and cluster response vectors based on their semantic similarity. Assuming that similar codes represent semantically similar responses, we propagate codes to responses in optimally homogeneous clusters. Cluster homogeneity is assessed by strategically querying informative responses and presenting them to a human rater. Following each manual coding, the method estimates the code distribution respecting a certainty interval and assumes a homogeneous distribution if certainty exceeds a predefined threshold. If a cluster is determined to certainly comprise homogeneous responses, all remaining responses are coded accordingly automatically. We evaluated the method in a simulation using different data sets.

Results: With an average miscoding of about 3%, the method reduced the manual coding effort by an average of about 52%.

Conclusion: Combining the advantages of automatic and manual coding produces considerable coding accuracy and reduces the required manual effort.

KEYWORDS
clustering, eco, effort reduction, exploring coding assistant, semi-automatic coding, support human raters

1 | INTRODUCTION

Evaluating open-ended text responses is known to be very demanding. It requires that manual coders not only deal with the cognitive task to comprehend and interpret text (Graesser & Kreuz, 1993; Kintsch & van Dijk, 1978; Perfetti & Joseph, 2014) but also that they use an evaluation scheme consistently to ensure reliable and valid coding. In this paper, we propose a method to support human coding through natural language processing.

A coding scheme is often created to account for different response scenarios, whereby coders can reliably code responses by matching them with possible answers contained within a set of reference texts. However, human coding decisions can be biased (Bejar, 2012) and objectivity compromised (see Klein & El, 2003). Despite these difficulties, open-ended

items are used in assessments due to the advantages they offer in comparison to multiple-choice tasks for certain constructs, including additional information beyond closed-format tasks (Frederiksen, 1984; Hancock, 1994; Millis et al., 2007). Open-ended tasks can demand the activation of higher-order cognitive skills (Hancock, 1994) or reveal whether testees profoundly understand a text (Rupp et al., 2016). Given their information-rich nature, open-ended formats are also used in questionnaires pertaining to fields such as medicine (e.g., Burla et al., 2008; Huston & Rowan, 1998) and self-regulation (e.g., Anthony et al., 2013).

Human raters are typically considered the gold standard for coding open-ended text responses because they can provide the necessary understanding to comprehend, interpret, and, ultimately, rate such texts. Nonetheless, likely because of their potential to save time and effort, as well as to improve coding consistency, researchers have increasingly developed automatic coding methods (see Burrows et al., 2015, for a somewhat outdated overview), which, in comparison to humans, are incapable of deeply understanding texts. Instead, these methods can code texts automatically by recognizing patterns. Various systems have already been developed to automatically and semi-automatically code text responses to assessment tasks (see Basu et al., 2013; Horbach et al., 2014; Leacock & Chodorow, 2003; Mieskes & Pado, 2018; Zehner et al., 2016; Zesch, 2015). While fully automated methods require a complete pre-coded data set to train a model, partially automated systems only code a subset of responses in a dataset automatically, with other responses coded manually. With few exceptions (e.g. Cai et al., 2019; Horbach & Pinkal, 2018), these methods are mostly static, meaning that a fixed amount of manually coded data is defined as training data to initiate the automatic coding process, but there is no dynamic interaction with human coders.

To code responses automatically, the system must collect information about the response universe, including the relevant semantics and the codes assigned. However, it remains unsolved how much information is required to make a coding decision with a minimal degree of certainty. This produces what is commonly described as an exploration-exploitation dilemma.

More formally, this refers to the general decision problem of whether (i) to apply an action from a sequence of finite actions for choosing the best currently available option given the current information (i.e., the option known to provide the greatest benefit for exploitation) or (ii) to explore new options that may provide greater benefit than the best currently known option. If no new option provides additional benefit, the latter action represents a waste of resources. This fundamental decision problem affects both humans (Wilson et al., 2014) and machines, especially in reinforcement learning (Kaelbling et al., 1996). Translated to the present paper's focus, this corresponds to the question: How many responses must be coded manually in order to obtain enough information for coding other responses automatically (with a high level of certainty), and which responses provide sufficient information in order to require the smallest possible number of manually coded responses?

2 | RESEARCH GOALS

To answer this question, we have developed *exploring coding assistant* (eco)—a semi-automatic method—that interactively and dynamically supports human coders in the manual coding process. The mechanism recognizes responses that can be coded automatically with a high degree of certainty and codes them in the background. These responses demonstrate substantial semantic similarity to responses that have already been coded manually, reducing the effort required from human coders. The assistant systematically queries responses for manual coding, which iteratively generates a training data set that enables the automatic assistant to train a coding model while automatically coding other responses in the background. The decision to execute the automatic coding process is made upon reaching a minimum level of statistical certainty. Thus, the method does not require a pre-coded data set and can be applied on the fly during the manual coding process. To evaluate the method, we simulated it using various pre-labelled data sets (including partial credit scores).

3 | SEMI-AUTOMATIC METHOD FOR CODING TEXT RESPONSES

This section describes the different steps involved in the proposed semi-automatic coding process with the aim of supporting the human rater. Figure 1 illustrates the process. Based on a pre-trained vector representation of semantics, (I), the open-ended text

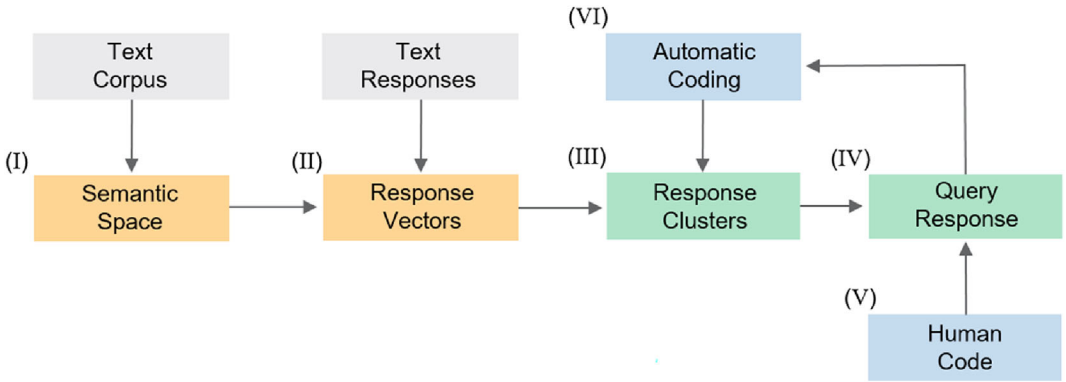

FIGURE 1 Flowchart of the semi-automatic coding procedure

responses are mapped into a semantic space across *n* dimensions (II). The response vectors are clustered based on their semantic similarity (III), implying the following assumptions:

1. Similar responses that demonstrate considerable semantic similarity will be assigned to the same cluster, and
2. semantically similar responses can be assigned to the same code.

As a result from these assumptions, the next task is to explore the clusters regarding the responses' code distribution and check for their homogeneity.

Assumptions are verified for each cluster by systematically querying responses from a cluster (IV) and presenting them to a human in the loop (V). Thereupon, the code distribution in the cluster is estimated using a critical interval. If the code distribution in a cluster is estimated to be homogeneous according to a predefined certainty threshold, all remaining responses in the cluster are automatically coded (VI) with the cluster's dominant code. Responses from heterogeneous clusters continue to be coded manually. Thus, the method supports the human rater during the process if the condition of high similarities between the responses leading to homogeneous clusters, allows it. The next sections of this paper detail the individual steps.

3.1 | Building a semantic space

To apply quantitative methods to texts that initially demonstrate qualitative properties in the form of unstructured information, these properties must be converted into quantitative properties. Semantic spaces enable numerical representations of a word's semantics and allow for mathematical calculations regarding their meaning (Deerwester et al., 1990; see Mikolov, Le, et al., 2013; Pennington et al., 2014). Semantic spaces represent words as *n*-dimensional vectors. A word's original semantics cannot be directly reconstructed with the vector representation but can be measured indirectly via the cosine similarity to another word vector.

The similarity of two vectors $sim\left(\vec{a},\vec{b}\right)$ with *n* dimensions can be measured by their angle θ, where words with a similar meaning—for example, synonyms—showing a small angle. The number of dimensions is often specified to be 300 (see Landauer et al., 1998; Mikolov, Chen, et al., 2013; Pennington et al., 2014).

$$sim\left(\vec{a},\vec{b}\right)=\cos(\theta)=\frac{\vec{a}\cdot\vec{b}}{\left\|\vec{a}\right\|\left\|\vec{b}\right\|}=\frac{\sum_{i=1}^{n}a_i b_i}{\sqrt{\sum_{i=1}^{n}a_i^2}\sqrt{\sum_{i=1}^{n}b_i^2}}. \quad (1)$$

The cosine similarity can take values in a range between $-1 < cos(\theta) < 1$. A similarity of $cos(\theta)=1$ indicates identical semantics of two words, for instance, synonyms, and a similarity of $cos(\theta)=0$ indicates semantically dissimilar words.

In recent decades, different methods for building semantic spaces have been established (see Deerwester et al., 1990; Devlin et al., 2019; Mikolov, Chen, et al., 2013; Pennington et al., 2014). In general, these processes are based on the distribution of words and the assumption that words in a similar context also share a similar meaning. With respect to the distributional hypothesis, words are considered similar when they demonstrate a similar distribution: "The distribution of an element will be understood as the sum of all its environments" (Harris, 1954, p. 146). From a technical perspective, a word's environment or context is determined by its co-occurrences, which can include all words in the document (Deerwester et al., 1990) or all words in a particular window around the target word (Mikolov, Chen, et al., 2013). In the context of newer methods, they are also known as word embeddings.

For the simulations in the present study, we used semantic spaces, built with word2vec (Mikolov, Chen, et al., 2013), which does not consider word order, minimizing syntactic variance. The German vector space model was trained with a sample of 500,000 documents from the German Wikipedia using a window of five words, considering the two words both before and after the target word. word2vec is based on neural nets and can be distinguished as two separate models, which iteratively train weights from the input. While the Continuous Bag of Words Model is trained to predict the target word based on context. Skip-gram is used to predict the context based on the target word. The resulting parameters from the model are used as word embeddings or semantic spaces. We extracted the weights of the Skip-gram model to represent the word vectors in 300 dimensions. The word embedding comprised 1,497,302 types (i.e., unique words and numbers).

For the datasets, including English text, we used a publicly available pre-trained model in English featuring 100 dimensions and a window of five words (Yamada et al., 2020). We chose a model with fewer dimensions to show that *eco* works under different conditions, including a different language as well as another dimensionality, to test minimal requirements. Furthermore, the number of dimensions plays a subordinate role when aggregating multiple vectors (Figure 2b) because information about single words is negligible. When measuring the relationship between two word representations, semantic spaces with more dimensions are often more accurate (Mikolov, Chen, et al., 2013).

3.2 | Normalization

The texts used to build the semantic space and the text responses were normalized by tokenization, lowercasing, and punctuation removal. Pre-processing text is common in many natural language processing methods and offers the advantage of reducing linguistic variability in texts. These steps are illustrated in Figure 2a.

3.3 | Computing response vectors

Given word embeddings, represent single words, it is necessary to aggregate word vectors to obtain a response vector by averaging all word vectors as a *bag-of-words* (Figure 2b). This means that syntactical order is ignored.

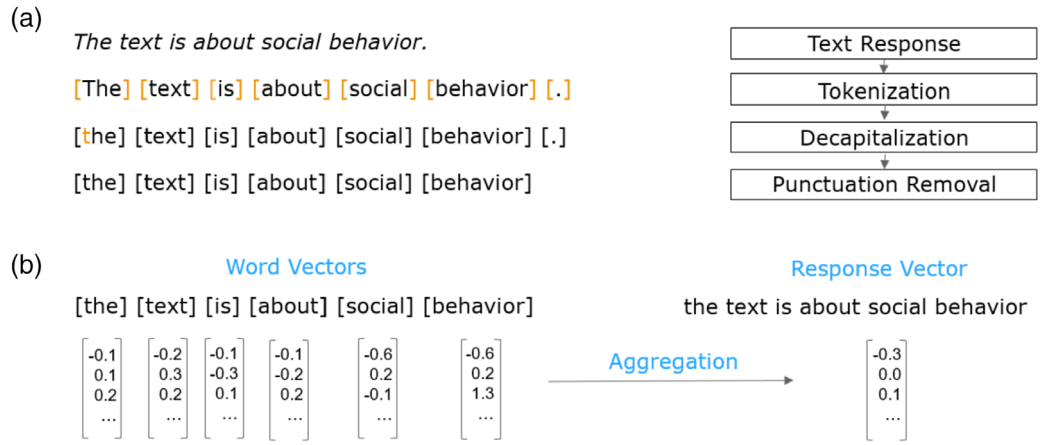

FIGURE 2 Pre-processing and vector representation of a text response

3.4 | Clustering

Text responses refer to their item. Since all students are responding to the same task, the responses should share similarities regarding the content. With this assumption in place, we cluster the text responses using their vector representations via hierarchical clustering.

Documents or texts can be grouped based on their semantics (see Aggarwal & Zhai, 2012; Willett, 1988). In the large-scale context, hierarchical clustering using cosine-based distance matrices and Ward's method as linkage criterion (Ward Jr, 1963; Zehner et al., 2016), or *k*-Means (Zesch, 2015) has proven to be effective. While *k*-means features a stochastic term that can provide different results with repeated use, hierarchical methods deterministically reproduce the same results depending on the selected linkage criterion, which determines how data points are merged into clusters. Centroid-based clustering methods offer the advantage of a centroid vector that can be generated by aggregating all response vectors. The response with the highest cosine similarity to the centroid vector represents the cluster's prototypical response. To use this information systematically, we have chosen a non-stochastic centroid-based hierarchical clustering method with Ward's method (Ward Jr, 1963).

3.5 | Explore versus exploit

Semantically similar responses should be assigned to the same code and form an optimally homogeneous cluster with respect to their codes. Given that the proper code distributions in the clusters are unknown, the assumption of homogeneity must be tested for each cluster. Only the coding of one human rater is required here.

The human rater is successively presented with responses from a given cluster, which allows us to estimate the code distributions (and their homogeneity) by examining a fraction of responses, the necessary amount of responses to exceed the certainty threshold, considering a critical interval. If a predefined certainty threshold T is reached, we anticipate that the whole cluster distribution is optimally homogeneous and, accordingly, code the remaining responses automatically.

3.6 | Uncertainty and confidence

In order to answer the questions of how many text responses inside a cluster are expected to be labelled as, for example, *correct* or *incorrect*, and how high the certainty is that the sampled distribution generalizes to all responses inside the cluster appropriately, the cluster's code distribution is estimated using a normal approximated critical interval (Cochran, 1991).

The number of responses with a particular code X is estimated by the number of the queried responses containing code X, x, and the number of all queried responses n.

$$p(X) = \frac{x}{n}. \tag{2}$$

The critical interval is defined by

$$p(X) \pm \left[t \sqrt{1 - \frac{n}{N}} \sqrt{\frac{pq}{n-1}} + \frac{1}{2n} \right], \tag{3}$$

where t indicates the normal deviate ($t = 1.96$, with a type-I-risk of $\alpha = .05$), N indicates the number of all responses in a cluster and n indicates the number of queried responses, whereby $p = x/n$ and $q = 1 - p$. The continuity correction is defined by $\frac{1}{2n}$ (see Cochran, 1991). The critical interval (CI) decreases steadily with each coded response until all responses from a cluster are coded manually. Thus, the certainty is represented by the lower bound of the CI. If certainty exceeds the predefined threshold T, we can assume that the entire cluster is optimally homogeneous and propagate automatic coding. The effort reduction, ER, is operationalized by the number of automatic codes in relation to the number of responses in a dataset.

To see how this works, it is worth considering some examples:

Scenario A: A cluster comprises $N = 20$ responses, coded as correct, with a perfectly homogeneous code distribution. A total of $x = 5$ answers are queried and presented to the human rater, who codes the responses as correct. The estimated proportion of correct responses in the cluster is $p(\text{correct}) = 1.00; 95\%\text{CI}\,[.90, 1.00]$. Because the lower confidence level does not exceed the predefined threshold of $T = 0.90$, the cluster is further explored. After querying and coding another correct response, the proportion of correct responses is estimated with $p(\text{correct}) = 1.00; 95\%\text{CI}\,[.92, 1.00]$. Given that the threshold is now exceeded, the assumption that the cluster is optimally homogeneous can be confirmed, and all additional 14 responses in the cluster are automatically coded according to the dominant coding, in this case, *correct*, saving $ER = 70\%$ of the coding effort.

Scenario B: Seven responses are queried from a heterogeneous cluster featuring a total of 20 responses. Four of the seven responses are coded as correct. The proportion of responses in the cluster is estimated to be $p(\text{correct}) = 0.57; 95\%\text{CI}\,[0.18, 0.96]$ and $p(\text{incorrect}) = 0.43; 95\%\text{CI}\,[0.04, 0.82]$. The estimated code distribution indicates insufficient homogeneity for automatic coding. Exploration will continue until the certainty value is exceeded or all responses have been coded manually.

The definition of T also determines the minimum number of codes required to exceed the threshold. With a lower T, the required number of manual codings decrease, which means that the estimates are potentially more error-prone.

For dichotomous coding, the distribution of only one code needs to be estimated since the other shows the reverse distribution. Yet, for technical reasons, each code distribution is estimated so that the method can be extended to multinomial and ordinal scores with more than two codes, which require all coding distributions to be estimated.

3.7 | Estimating the number of clusters

We aim for relatively large clusters because we want to reduce as much coding effort as possible. Cluster homogeneity increases as the number of clusters increases. They demonstrate perfect homogeneity if the number of clusters corresponds to the number of responses because each response is assigned to its own cluster, which precludes an automatic coding process. Consequently, we must solve the optimization problem of there being two conflicting target functions. That is, clusters should reach the smallest possible size to achieve optimal homogeneity, but they also should be sufficiently large to enable exploration.

There is no perfect solution to this problem because it depends on the user's expectations of the effort reduction and the accepted miscoding rate. If users want to reduce coding effort as much as possible, a small k is needed, more likely resulting in larger and more heterogeneous clusters and increasing the number of possible incorrect codings.

Additionally, the threshold value relates directly to the effort reduction expectation. Even in a maximally homogeneous cluster, a minimum number of manually coded responses are required to exceed the threshold.

In Scenario A, six responses must be queried from a perfectly homogeneous cluster of 20 responses to exceed the threshold of $T = .90$, allowing a maximum of 14 responses to be coded automatically. Meanwhile, a threshold of $T = .50$ requires only three manual codings to exceed the certainty threshold for a homogeneous cluster. Although this reduces the overall coding effort, it may also generate more incorrect codings because smaller sample sizes frequently result in heterogeneous clusters being incorrectly identified as homogeneous.

Using the hierarchical cluster structure and assuming a certain value of k enables determination of the maximum effort reduction for each k in the range $1 \le k \le N$, assuming that all clusters demonstrate perfect homogeneity. Thus, the required k can be determined by working backwards.

For varying items and the same expected effort reduction, k differs because the item responses vary in complexity. This is reflected in different hierarchical cluster structures. By determining k backwards, we do not have to set a fixed generic value of k; instead, we can set it as a function of the effort reduction expectation.

3.8 | Query strategies

Systematically sampling data can reduce the amount of data needed to train a model while simultaneously increasing the model's performance (see Lewis & Gale, 1994; Settles, 2009), as has been demonstrated in the context of, for example, short answer scoring (Horbach & Palmer, 2016). Moreover, during unsupervised learning, patterns of response characteristics and their unseen ratings can become visible. For example, if certain responses are grouped into a cluster, the responses share similar attributes, represented by the prototypical response, with the highest cosine similarity to the centroid vector. Thus, the similarity of the response vector to the centroid vector contains similar information to the word frequency (see Salton & Buckley, 1988) which also affects the similarity. If a response contains various words frequently used in the cluster, it shows a smaller distance to the centroid. This information enables the development of specifying strategies for sampling responses in a semantic space throughout the process to identify the most informative response in the cluster. In this context, the most relevant information is that which allows for earlier recognition of whether or not a cluster is heterogeneous, avoiding miscoding. Given the prototypical response could be a misleading indicator for this purpose, we tested three different sampling strategies: (1) a random-based strategy and two distance-based strategies, (2) with preferential querying of prototypical responses, *near centroid*, and (3) preferential querying of atypical responses, *far from centroid*.

3.9 | Data sets

We employed different data sets to tune and evaluate the method. The data from the 2012 programme for international student assessment (PISA) assessment (OECD, 2013) were used solely to optimize

TABLE 1 Data sets

Data set	Items	*n*	Tokens (median)	Raters	Language	Usage
PISA '12	10	37,072	12	1	German	Train
PISA '15 access and retrieve	7	7681	9	1	German	Test
PISA '15 integrate and interpret	14	14,461	13	1	German	Test
PISA '15 reflect and evaluate	17	17,069	19	1	German	Test
Powergrading	10	6980	3	3	English	Test
ASAP	10	17,207	40	2	English	Test

Note: The table shows the data sets used to evaluate the method with their corresponding numbers of items and total responses *n*, the median number of tokens (i.e., words and numbers) in the responses, the number of raters involved in the coding process, the data set language and if it was used for hyperparameter tuning or evaluation.
Abbreviations: ASAP, automated student assessment prize; PISA, programme for international student assessment.

the method's parameters. Next, we describe the method. Finally, we report simulations for evaluating the method and chosen parameter values, for which the other three data sets were used.

3.9.1 | Programme for international student assessment 2012

For hyperparameter optimization (i.e., to estimate the optimal number of clusters, the best sampling strategy and the best threshold for automatic coding), we used 10 items from the German PISA 2012 (OECD, 2013), which was previously used for automatic coding (Zehner et al., 2016). These items represent a cross-sectional measurement of various competencies (reading, science and mathematics; see Zehner et al., 2016), which is reflected in the variation in response characteristics. This international assessment is conducted every 3 years and tests 15-year old students in mathematics, science, and reading literacy. From the selected 10 items, eight items assess reading, one item assesses mathematical, and one item assesses science literacy. All 10 items are coded dichotomously as correct/incorrect. Assessment data typically contains empty responses. For the purpose of comparability empty responses were removed.

3.9.2 | Programme for international student assessment 2015

This data set includes German short text responses from PISA 2015 (OECD, 2017) stemming from 15-year-old students from German schools. The 38 reading literacy items are divided into three distinct item types with different characteristics in terms of, for example, complexity and median response length (Table 1). Access and Retrieve tasks are typically short items. In most cases, this means that parts of the text have to be reproduced from the read text (e.g., recall animal species from a given text; Andersen & Zehner, 2021). In integrate and interpret tasks, components of the text must be combined or interpreted, and reflect and evaluate tasks require that students must reflect beyond the text base. Some items feature dichotomous coding assignments (correct/incorrect), and some feature three-level ordinal scaled scoring. The advantages of data from established large-scale assessments are that the measurement instruments have been tested and optimized using various high-quality criteria (Berliner, 2020). We also removed empty responses.

3.9.3 | Powergrading

This English-language data set collected by (Basu et al., 2013) includes 10 items, resulting in a total of $n = 6980$ short responses. The items include questions from the United States Citizenship Exam. Responses to these questions were collected via the crowdsourcing platform Amazon Mechanical Turk. Responses are comparatively short and evoke very low-linguistic variance, comparable to items from the PISA access and retrieve item type, in terms of linguistic aspects. However, items do not contain reading texts with information. Three raters conducted dichotomous coding.

3.9.4 | Automated student assessment prize

The data set of 10 items, also in English, was made publicly available during a competition on the data science platform Kaggle. The automated student assessment prize (ASAP; Kaggle, 2012) was intended to encourage the development of new scoring methods. The responses are longer than those pertaining to the other data sets. The items targeted critical thinking, and the ordinally scaled codes included three or four scores points per item. Each item was scored by two raters.

4 | SIMULATIONS

As the target criteria of the presented method, we report the percentage of reduced effort, the agreement between human and machine for automatically coded responses in terms of quadratic weighted kappa (Cohen, 1960, 1968; Fleiss, 1971) and the total accuracy of all responses to determine the effective coding accuracy of the method. The general accuracy includes the proportions of both human coding and automatic coding and represents, in relation to the effort

reduction *ER*, the overall success of the method. It is important to note here that all manually coded responses are considered to be coded correctly, which corresponds to a general accuracy of 100% for a fully manually coded dataset.

Additionally, for the PG and ASAP datasets, which were coded by multiple human raters, the human raters' agreement was reported as the human–human agreement κ_{HH}. As one simulation was performed for each rater, in addition, the agreement κ_{HS} between one automatically supported human coder and the other human coders was calculated as well as the agreement κ_{SS} between all raters that were supported by an automatic assistant. Kappa can have values between $-1 \leq \kappa \leq 1$, while a value of $\kappa = 1$ means perfect coder agreement, and a value of $\kappa = 0$ describes random code assignment in terms of the prior distribution (see Landis & Koch, 1977, for a more detailed segmentation).

4.1 | Simulation for hyperparameter optimization using German PISA 2012 data

For the semi-automatic method, two important parameters must be defined in advance. First, there is the optimal number of clusters *k* for grouping the response vectors, and second, the certainty threshold that must be reached for automatic coding to be executed. Additionally, as the query strategies have not been compared empirically yet, the strategy had to be defined as well. Since we could not expect these variables to be fully independent, we performed grid search to find the optimal combination of hyperparameters, in which every possible combination of parameter values was simulated for each item. The results were then averaged across items since we wanted to test the parameters' suitability for a generic application of the method, independent of specific items and response characteristics. In practical use, grid search could be applied to any labelled dataset for tailoring the hyperparameter values.

A simulation of the method was performed on the German PISA 2012 data. For each of the 10 items, six certainty thresholds $T \epsilon \{0.70, 0.75, 0.80, 0.85, 0.09, 0.95\}$ and eight levels of expected maximum effort reduction $MER \epsilon \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%\}$ were tested to find the optimal *k* value. Furthermore, three different query strategies were tested (*random*, *near centroid*, *far from centroid*). This way, we performed a total of 1620 simulations to find the parameters that allowed the optimal application across multiple items. So, for each of the 10 items, this corresponded to 162 results, and there were 10 results for each parameter combination, which were aggregated across items for choosing optimal generic hyperparameters. For doing so, we measured the effort reduction and automatic coding agreement as optimization criteria.

4.2 | Simulations for evaluating the semi-automatic coding method on other data sets

To select the optimal parameter values, the 162 results were sifted and reduced step by step on the basis of exploratively defined criteria. After an initial review of the data, a minimum reached effort reduction was set to $ER > 50\%$, which was the case for 38 of the 162 results. We took the case with the highest automatic coding agreement. Thus, the parameters were defined with $T = 0.85$, $MER = 0.80$, and a query strategy preferring responses with the highest distance to the cluster centroid, which results in an average general accuracy of 96.83% and an average coding effort reduction of 53.27%. The difference between the maximum and the real effort reduction is due to the fact that the *MER* requires perfect homogeneous clusters, which are rarely present in real-world data.

Effort reduction and general accuracy depend on the respective parameters and the item complexity. For example, the effort reduction takes a wide range between $2.03\% < ER < 89.07\%$, while the general accuracy ranges between $83.66\% < Accuracy < 100\%$ regarding all simulation results.

The specified parameters were then used to simulate the support provided by the automatic assistant to the rater in the other three datasets. To that end, a simulation was performed for each rater per item. All results are presented in detail for each item in Appendix A (Tables A1–A5).

The method provides different effort reductions and accuracies (Table 2). The largest effort reduction was achieved for the PISA data with the item type *access and retrieve*. Here, a total of 66.98% of the effort could be saved on average, whereby the range covers a similar area as with the PG items. There, 39.11% of the effort could be saved for the poorest performing item and 80.23% for the highest performing item. In the data set with the longest responses and polytomous coding (ASAP), only 19.63% of the coding effort could be saved on average, with no automatic coding performed on one item, resulting

TABLE 2 Effort reduction and accuracy

		Effort reduction (*ER*) in %			General accuracy in %		
Data set	*n*	Mean	Min	Max	Mean	Min	Max
PISA '15 access and retrieve	7	66.98	37.85	79.86	97.88	94.77	100.00
PISA '15 integrate and interpret	14	58.73	21.65	80.04	97.30	92.98	99.55
PISA '15 reflect and evaluate	17	40.04	15.26	71.00	94.47	90.42	96.93
Powergrading	10	73.57	39.11	80.23	99.12	95.70	100.00
ASAP	10	19.63	0.00	53.09	95.91	91.94	100.00

Note: The table shows the number of items in the data set (*n*), the effort reductions (*ER*) and general accuracy (including manual and automatic codings).

in 100% general accuracy. In the power grading dataset, responses of three items were also coded 100% correctly (items 1/3/5; see Appendix), although more than 79% were machine coded for all three.

To investigate whether the different results, general accuracy and effort reduction, were statistically related to the data set, an ANOVA was calculated for both measurements. The effort reductions differed significantly by data set $F(4, 83) = 48.34, p < 0.001, \eta^2 = 0.70$ as did the general accuracy, $F(4, 83) = 22.71, p < 0.001, \eta^2 = 0.52$, suggesting item- and sample-related differences.

5 | DISCUSSION

Depending on the item type and the related responses, *eco* was able to reduce a large proportion of coding effort, with only a marginal number of miscodes on average. For items with poorer automatic coding, the automatic assistant only accepts a limited number of errors if the distribution is heterogeneous (see ASAP, Item 1), leading to further manual codings or even a complete manual coding process (see ASAP, Item 2). The differences in effort savings and accuracies differ statistically between the data sets, with the difference in general accuracy in a clear range between 90 and 100%, showing that the method accurately estimates the certainty marker for potential automatic codings. For item 2 (ASAP), no response was automatically coded, as there was insufficient statistical confidence. This demonstrates the importance of accuracy over effort reduction. The responses of this item contain the longest responses on average with four coding levels. Short responses and, therefore, the least complex ones, were coded best (see PG, Item 4). That means these contain the smallest number of miscoding and the largest proportion of effort reduction. Whereby the complexity can be considered low not only due to the short length of responses but also because of the few coding levels (i.e., dichotomous scores). Accordingly, we identified two main problems that complicated the automatic coding of responses: a higher number of coding levels and long text responses.

The use of several coding levels create more (spatial) boundaries between the responses in the hyperdimensional semantic space, making it more difficult to distinguish between them during the clustering process. As can be seen from the results, this was not only a potential source of error for automatic coding but also human coding. The human coding of the critical item 2 (ASAP), for example, had an inter-rater reliability of only $IRR = 0.80$. For item 3 (ASAP), the agreement was even lower with $IRR = 0.60$, which was also reflected in the low-performance rate of the automatic coding due to low-statistical certainty. In contrast, for example, item 1 (PG), which contains only three tokens per response (median), shows a 100% total coding agreement with $ER = 80.09\%$. The *ER* even exceeds the *MER*, because of its high number of identical responses. Related to this was the content of the responses, which affected their representation in the semantic space and thus also their assignment to the respective clusters. This suggests an influence of longer responses as they may have a different ratio of relevant to irrelevant words. The influence of linguistic features on coding success can be covered by further research and requires further analysis. In the following, we discuss known issues and how they influence coding success for text responses.

One factor that particularly affects longer responses is the use of bag-of-words and the resulting loss of syntactic information. Although most information is represented by semantics (Landauer, 2002), an optimization that also considers word order might produce more accurate patterns but requires more training examples due to the increasing linguistic variety of the text and the number of word combination sto respond to a particular task. Especially syntactical organization differs across languages (Evans & Levinson, 2009). Another implied restriction concerns negations. By neglecting sentence structure, a negation loses the reference to the relevant word. For example, the response, 'He did not tell her to keep the secret' would be treated in the same way as the response, 'He did tell her not to keep the secret.' Although both sentences contain the same words, the differential use of negation significantly changes the sentence's meaning. This difference in meaning can be relativized in the assessment context because a specific condition is not usually described contrarily using a negation.

The method developed was designed for nominally coded short responses. However, the method has been demonstrated to work with ordinally scaled codes that are methodologically downscaled to a nominal scale. A typical scoring scheme for ordinal-scaled items is that the naming of a particular term is scored with one point while a naming with an additional argument is scored with two points, although the level of detail of the argument can also be subject to an additional gradation. Thus, the responses contain different intersections of information. A decomposition of the response into different information units or *n*-grams combined with a rule-learner could better distinguish between the graduated differences and detect essential sub-sentence structures in long responses. As the decomposition of responses produces further problems, additional techniques would need to be considered. Consider, for example, a decomposition of the response 'John was worried about his dog. He hasn't seen it for a while.' The pronoun (*he*) in the second sentence is referring to the name *John* in the first sentence. After a sentence decomposition, the information concerning the reference would be lost (see Bexte et al., 2021; Mitkov, 2002). An additional focus on coreferences could mitigate this information loss by referring pronouns to the relevant entity, meaning that additional rules (e.g. Hobbs, 1978) or transformers (e.g. Joshi et al., 2020) could be used to reconstruct this coreference automatically.

The assistant crudely balances effort reduction against potential miscodings by predefining parameters such as the number of clusters and the certainty threshold. Pre-processing steps may have additionally impacted the weighting of the two target variables. For example, stemming (reducing a word to its stem) reduces the linguistic variability of a text, promoting generalizability to prevent overfitting but removes morphological information. The effect of different pre-processing steps on the results of coding should be examined in more detail.

Applying the method in an authentic assessment scenario also provides various design options. For example, implementing the method in an environment with a user interface (e.g., Andersen & Zehner, 2021)

allows for the inclusion of additional concepts, such as how responses are displayed to the rater, whether as single responses or as multiple responses featuring a code suggestion requiring the rater's approval. Furthermore, the decision to perform automatic coding depends on whether a certain estimated proportion of codes exceeds the certainty threshold. This responsibility could be given to the human rater to decrease the control of the assistant and allow for increased flexibility.

Human ratings are considered the gold standard and used as a benchmark for the evaluation of automatic coding approaches (see Shaw et al., 2020). In applications where human raters are integrated into a loop, like in the presented method, human raters are sometimes also called *Oracle*, which trivializes human coding errors at least from a philosophical, if not even from an empirical point of view. However, codings can also vary between human raters, which should be taken into account when evaluating an automatic coding system. Data sets are often only rated by one rater, and that coding serves to train and test the method simultaneously, with the predicted coding compared to the true coding. Limited agreement between the two values can indicate a failed coding strategy or the failure to code a critical item. Automatic and semi-automatic methods can be used to detect these systematic coding difficulties and revise items and coding guidelines. To measure a coding system, two independent human ratings are indispensable because they allow comparisons between automatic and single-human coding and codings by two or more human coders.

Language diversity is critical for a broad multilingual application. Using the same method in different languages can lead to different results because aspects such as pre-processing affect the responses differently. Important factors could include response length and variance (Horbach & Zesch, 2019). Long responses usually contain more semantic information but can also contain more noise, that is, words that do not contain coding-relevant information. Linguistic variability (see Horbach & Zesch, 2019) can affect a response's vector representation, producing heterogeneous clusters and degrading the coding system. Languages also differ in their information densities (Coupé et al., 2019; Pellegrino et al., 2011), meaning different languages use different numbers of syllables to communicate the same amount of information. Normalization steps, such as stemming or removing stop words (semantically irrelevant words), could be used to develop a cross-language method and minimize linguistic variability. This involves providing stop words in predefined lists of varying length, depending on the target language. If a method is planned to be applied across languages, sufficient testing is critical because comparing results, especially at the international level, as in the case of PISA, can impact future policy decisions (Ertl, 2006).

6 | CONCLUSION

For practical implementation, *eco* was designed to be applicable as simply as possible. The assistant cooperates with the human rater to reduce coding effort. By pre-selecting responses for the rater, *eco* uses the continuously gathered coding information to estimate the chance of a successful automatic coding process and applies automatic coding if the certainty exceeds a predefined threshold.

After listing potential issues and features that were not considered in this basic approach, improvements can be expected, particularly regarding coding accuracy, where the method will consider more detailed linguistic features. The hyperparameter selection can be limited to the choice of certainty threshold and the number of clusters, which simplifies a practical realization. It could be shown that with general parameter settings (see simulation two), sufficient results can be achieved over several tasks regarding effort reduction and the accuracy of the automatic coding. The dependence of the cluster number on the threshold and the use of a general clustering method (across all items) were tested and can be assumed as a default value for an application with the potential for individual optimization. Optimization of the parameters based on older datasets of a specific item to train the assistant for the coding of a new dataset of the same item is also conceivable and could lead to better results.

ACKNOWLEDGMENT

Open Access funding enabled and organized by Projekt DEAL.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/jcal.12717.

DATA AVAILABILITY STATEMENT

Due to repeated measurements in PISA, item contents are confidential. Text response data, which reveals those contents, can thus only be requested from the OECD or respective PISA National Centers. Other data than PISA data used in this study is cited accordingly and can be accessed via the original reference.

ORCID

Nico Andersen https://orcid.org/0000-0002-8333-9071
Fabian Zehner https://orcid.org/0000-0003-3512-1403
Frank Goldhammer https://orcid.org/0000-0003-0289-9534

REFERENCES

Aggarwal, C. C., & Zhai, C. (2012). A survey of text clustering algorithms. In C. C. Aggarwal & C. Zhai (Eds.), *Mining text data* (pp. 77–128). Springer.

Andersen, N., & Zehner, F. (2021). shinyReCoR: A shiny application for automatically coding text responses using R. *Psychology*, *3*(3), 422–446.

Anthony, J. S., Clayton, K. E., & Zusho, A. (2013). An investigation of students' self-regulated learning strategies: Students' qualitative and quantitative accounts of their learning strategies. *Journal of Cognitive Education and Psychology*, *12*(3), 359–373. https://doi.org/10.1891/1945-8959.12.3.359

Basu, S., Jacobs, C., & Vanderwende, L. (2013). Powergrading. A clustering approach to amplify human effort for short answer grading. In D. Lin & M. Collins (Eds.), *Transactions of the Association for Computational Linguistics* (Vol. 1, pp. 391–402). MIT Press.

Bejar, I. I. (2012). Rater cognition. Implications for validity. *Educational Measurement: Issues and Practice*, *31*(3), 2–9.

Berliner, D. C. (2020). The implications of understanding that PISA is simply another standardized achievement test. In G. Fan & T. S. Popkewitz (Eds.), *Handbook of education policy studies: School/university, curriculum, and assessment* (Vol. 2, pp. 239–258). Springer Singapore.

Bexte, M., Horbach, A., & Zesch, T. (2021). Implicit Phenomena in Short-answer Scoring. Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language, 11–19.

Burla, L., Knierim, B., Barth, J., Liewald, K., Duetz, M., & Abel, T. (2008). From text to codings: Intercoder reliability assessment in qualitative content analysis. *Nursing Research*, *57*(2), 113–117. https://doi.org/10.1097/01.NNR.0000313482.33917.7d

Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, *25*(1), 60–117. https://doi.org/10.1007/s40593-014-0026-8

Cai, Z., Siebert-Evenstone, A., Eagan, B., Shaffer, D. W., Hu, X., & Graesser, A. C. (2019). nCoder+: A semantic tool for improving recall of nCoder coding. In *Advances in quantitative ethnography* (Vol. 1112, pp. 41–54). Springer.

Cochran, W. G. (1991). *Sampling techniques* (3rd ed.). John Wiley & Sons.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213–220.

Coupé, C., Oh, Y. M., Dediu, D., & Pellegrino, F. (2019). Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, *5*(9), eaaw2594. https://doi.org/10.1126/sciadv.aaw2594

Deerwester, S., Dumais, S. T., Furnas, G. W., & Landauer, T. K. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6\391::AID-ASI1.3.0.CO

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 (Vol. *1*, pp. 4171–4186).

Ertl, H. (2006). Educational standards and the changing discourse on education: The reception and consequences of the PISA study in Germany. *Oxford Review of Education*, *32*(5), 619–634.

Evans, N., & Levinson, S. C. (2009). The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and Brain Sciences*, *32*(5), 429–448. https://doi.org/10.1017/S0140525X0999094X

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*, 378–382.

Frederiksen, N. (1984). The real test bias: Influences of testing on teaching and learning. *American Psychologist*, *39*(3), 193–202. https://doi.org/10.1037/0003-066X.39.3.193

Graesser, A. C., & Kreuz, R. J. (1993). A theory of inference generation during text comprehension. *Discourse Processes*, *16*(1–2), 145–160. https://doi.org/10.1080/01638539309544833

Hancock, G. R. (1994). Cognitive complexity and the comparability of multiple-choice and constructed-response test formats. *The Journal of Experimental Education*, *62*(2), 143–157. https://doi.org/10.1080/00220973.1994.9943836

Harris, Z. S. (1954). Distributional structure. *WORD*, *10*(2–3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

Hobbs, J. R. (1978). Resolving pronoun references. *Lingua*, *44*(4), 311–338. https://doi.org/10.1016/0024-3841(78)90006-2

Horbach, A., & Palmer, A. (2016). Investigating Active Learning For Short-Answer Scoring, Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications.

Horbach, A., Palmer, A., & Wolska, M. (2014). Finding a Tradeoff Between Accuracy and Rater's Workload in Grading Clustered Short Answers, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).

Horbach, A., & Pinkal, M. (2018). Semi-supervised Clustering for Short Answer Scoring. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Horbach, A., & Zesch, T. (2019). The influence of variance in learner answers on automatic content scoring. *Frontiers in Education*, *4*, 28. https://doi.org/10.3389/feduc.2019.00028

Huston, P., & Rowan, M. (1998). Qualitative studies. Their role in medical research. *Canadian Family Physician*, *44*, 2453–2511.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. In M. Johnson, B. Roark, & A. Nenkova (Eds.), *Transactions of the Association for Computational Linguistics* (Vol. 8, pp. 64–77). MIT Press.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285---.

Kaggle. (2012). Automated Student Assessment Prize: Phase Two - Short Answer Scoring. https://www.kaggle.com/c/asap-sas/

Kintsch, W., & van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, *85*(5), 363–394.

Klein, J., & El, L. P. (2003). Impairment of teacher efficiency during extended sessions of test correction. *European Journal of Teacher Education*, *26*(3), 379–392. https://doi.org/10.1080/0261976032000128201

Landauer, T. K. (2002). In B. Ross (Ed.), *On the computational basis of learning and cognition: Arguments from LSA* (Vol. 41, pp. 43–84). Psychology of Learning and Motivation.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2–3), 259–284. https://doi.org/10.1080/01638539809545028

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*, 159–174.

Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, *37*, 389–405.

Lewis, D. D., & Gale, W. A. (1994). A Sequential Algorithm for Training Text Classifiers. SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and development in information retrieval (pp. 3–12)

Mieskes, M., & Pado, U. (2018). Work smart-reducing effort in short-answer grading. Proceedings of the 7th Workshop on NLP for Computer Assisted Language Learning (pp. 57–68)

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv*, 1301.3781.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv*, 1309.4168.

Millis, K., Magliano, J., Wiemer-Hastings, K., Todaro, S., & McNamara, D. (2007). Assessing and improving comprehension with latent semantic analysis. In T. K. Landauer, D. S. McNamara, S. Dennis, & K. Walter (Eds.), *Handbook of latent semantic analysis* (pp. 207–225). Psychology Press.

Mitkov, R. (2002). *Anaphora resolution*. Routledge.

OECD. (2013). *PISA 2012 assessment and analytical framework: Mathematics, reading, science, problem solving and financial literacy*. OECD Publishing.

OECD. (2017). *PISA 2015 assessment and analytical framework: Science, reading, mathematic, financial literacy and collaborative problem solving*. OECD Publishing.

Pellegrino, F., Coupé, C., & Marsico, E. (2011). A cross-language perspective on speech information rate. *Language*, *87*(3), 539–558. https://doi.org/10.1353/lan.2011.0057

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543.

Perfetti, C., & Joseph, A. S. (2014). Word knowledge in a theory of reading comprehension. *Scientific Studies of Reading*, *18*(1), 22–37. https://doi.org/10.1080/10888438.2013.827687

Rupp, A. A., Ferne, T., & Choi, H. (2016). How assessing reading comprehension with multiple-choice questions shapes the construct: A cognitive processing perspective. *Language Testing*, *23*(4), 441–474. https://doi.org/10.1191/0265532206lt337oa

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Settles, B. (2009). Active learning literature survey. University of Wisconsin-Madison, Department of Computer Sciences https://research.cs.wisc.edu/techreports/2009/TR1648.pdf

Shaw, D., Bolender, B., & Meisner, R. (2020). Quality control for automated scoring in large-scale assessment. In D. Yan, A. A. Rupp, & P. W. Foltz (Eds.), *Handbook of automated scoring* (pp. 241–262). Chapman and Hall/CRC.

Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.

Willett, P. (1988). Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management*, *24*(5), 577–597. https://doi.org/10.1016/0306-4573(88)90027-1

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., & Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, *143*(6), 2074–2081. https://doi.org/10.1037/a0038199

Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., & Matsumoto, Y. (2020). Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 23–30).

Zehner, F., Sälzer, C., & Goldhammer, F. (2016). Automatic coding of short text responses via clustering in educational assessment. *Educational and Psychological Measurement*, *76*(2), 280–303. https://doi.org/10.1177/0013164415590022

Zesch, T. (2015). Reducing Annotation Efforts in Supervised Short Answer Scoring. Proceedings of the Tenth Workshop on Innovative Use of Nlp for Building Educational Applications (pp. 124–132).

How to cite this article: Andersen, N., Zehner, F., & Goldhammer, F. (2023). Semi-automatic coding of open-ended text responses in large-scale assessments. *Journal of Computer Assisted Learning*, *39*(3), 841–854. https://doi.org/10.1111/jcal.12717

APPENDIX A

TABLE A1 PISA '15 access and retrieve simulation results

Item	*n*	Codes	Median tokens	Coverage in %	*ER* in %	General Acc. In %	κ_{AC}
R227Q06	1132	2	16	78.55	79.86	98.85	0.79
R420Q02	1144	2	8	96.34	77.62	98.95	0.89
R420Q09	1144	2	1	99.18	79.81	100.00	1.00
R442Q02	1005	2	11	86.18	69.35	96.72	0.81
R455Q03	1164	2	5	86.49	73.97	98.97	0.93
R460Q01	1059	2	19	87.44	50.42	96.88	0.77
R466Q02	1033	2	14	84.67	37.85	94.77	0.61

Note: The tables show the simulation results per item and rater, the relative coverage of the tokens in the semantic space and the proportion of automatically coded responses *ER*, the proportion of correctly coded responses (general accuracy; including manual and automatic codes), and the agreement between the automatically predicted codes and the true codes κ_{AC}.

TABLE A2 PISA '15 integrate and interpret simulation results

Item	*n*	Codes	Median tokens	Coverage in %	*ER* in %	General Acc. In %	κ_{AC}
R055Q03	1077	3	11	90.95	64.25	98.14	0.93
R055Q05	1040	2	15	91.59	53.85	96.54	0.47
R102Q04	926	2	18	89.87	69.11	97.95	0.94
R102Q05	1090	2	2	69.23	74.22	99.17	0.98
R406Q01	1063	2	17	92.41	60.77	96.8	0.67
R406Q02	910	2	13.5	91.17	21.65	98.57	0.58
R406Q05	1038	2	13	92.33	55.20	97.98	0.78
R412Q08	783	2	25	92.88	43.81	92.98	0.67
R420Q10	1033	3	23	85.08	67.96	97.48	0.74
R432Q01	1112	2	2	97.07	80.04	99.55	0.95
R437Q07	830	2	15	90.80	43.98	94.34	0.00[a]
R442Q03	960	2	13	82.45	49.58	97.40	0.73

Note: The tables show the simulation results per item and rater, the relative coverage of the tokens in the semantic space and the proportion of automatically coded responses *ER*, the proportion of correctly coded responses (general accuracy; including manual and automatic codes), and the agreement between the automatically predicted codes and the true codes κ_{AC}.

[a]Due to uniform automatic coding, the kappa value cannot objectively reflect the agreement, since kappa results in 0, even with high accuracy.

TABLE A3 PISA '15 reflect and evaluate simulation results

Item	*n*	Codes	Median tokens	Coverage in %	*ER* in %	General Acc. In %	κ_{AC}
R055Q02	996	2	20	90.71	45.88	92.87	0.40
R067Q04	1096	3	29	88.24	19.53	94.43	0.48
R067Q05	1089	3	30	90.20	34.16	95.04	0.34
R111Q02B	995	3	24	89.44	27.14	94.77	0.57
R111Q06	966	3	22	89.10	26.81	92.86	0.16
R219Q02	1045	2	20	88.58	56.65	95.31	0.22
R227Q03	1003	2	12	89.63	58.03	95.71	0.73
R404Q10A	977	2	25	88.59	45.96	92.32	0.62
R404Q10B	944	2	26	88.56	28.28	93.75	0.53
R420Q06	970	2	19	88.53	15.26	94.74	0.17
R432Q05	1021	2	16	90.49	48.29	94.61	0.46
R442Q05	1002	2	14	88.49	33.33	90.42	0.36
R442Q06	695	2	25	88.32	44.32	95.11	0.00[a]
R446Q06	1107	2	18	88.96	71.00	96.93	0.81
R453Q04	1026	2	20	91.82	34.99	95.91	0.25
R453Q06	1049	2	12	88.35	59.01	96.66	0.45
R455Q02	1088	2	13	92.08	32.08	94.49	0.24

Note: The table shows the simulation results per item and rater, the relative coverage of the tokens in the semantic space and the proportion of automatically coded responses *ER*, the proportion of correctly coded responses (general accuracy; including manual and automatic codes), and the agreement between the automatically predicted codes and the true codes κ_{AC}.

[a]Due to uniform automatic coding, the kappa value cannot objectively reflect the agreement, since kappa results in 0, even with high accuracy.

TABLE A4 Powergrading simulation results

Item	*n*	Codes	Median tokens	Cove-rage in %	Rater	κ_{HH}	κ_{HS}	κ_{SS}	ER in %	General Acc. In %	κ_{AC}
1	698	2	3	99.83	R1	0.99	0.99	0.99	80.09	100.00	0.00[a]
1	698	2	3	99.83	R2	0.99	0.99	0.99	80.09	100.00	0.00[a]
1	698	2	3	99.83	R3	0.99	0.99	0.99	80.09	100.00	0.00[a]
2	698	2	3	99.73	R1	0.95	0.95	0.95	79.94	100.00	1.00
2	698	2	3	99.73	R2	0.95	0.95	0.95	80.23	100.00	1.00
2	698	2	3	99.73	R3	0.95	0.95	0.95	79.66	100.00	1.00
3	698	2	7	99.35	R1	0.57	0.53	0.57	56.45	96.99	0.51
3	698	2	7	99.35	R2	0.57	0.52	0.57	52.44	96.42	0.46
3	698	2	7	99.35	R3	0.57	0.56	0.57	39.11	95.70	0.55
4	698	2	1	98.65	R1	0.86	0.85	0.86	72.92	99.14	0.92
4	698	2	1	98.65	R2	0.86	0.85	0.86	72.49	98.85	0.89
4	698	2	1	98.65	R3	0.86	0.85	0.86	68.62	99.00	0.92
5	698	2	1	97.60	R1	0.83	0.83	0.83	79.94	100.00	0.00[a]
5	698	2	1	97.60	R2	0.83	0.83	0.83	80.09	100.00	0.00[a]
5	698	2	1	97.60	R3	0.83	0.83	0.83	79.94	100.00	0.00[a]
6	698	2	1	99.34	R1	0.84	0.84	0.84	75.64	99.71	0.98
6	698	2	1	99.34	R2	0.84	0.84	0.84	78.37	99.86	0.99
6	698	2	1	99.34	R3	0.84	0.82	0.84	72.78	98.71	0.92
7	698	2	5	99.81	R1	0.85	0.86	0.85	73.64	98.71	0.52
7	698	2	5	99.81	R2	0.85	0.83	0.85	78.51	97.99	0.41
7	698	2	5	99.81	R3	0.85	0.84	0.85	73.93	99.14	0.62

(Continues)

TABLE A4 (Continued)

Item	n	Codes	Median tokens	Cove-rage in %	Rater	κ_{HH}	κ_{HS}	κ_{SS}	ER in %	General Acc. In %	κ_{AC}
8	698	2	4	99.53	R1	0.97	0.96	0.97	79.37	99.57	0.99
8	698	2	4	99.53	R2	0.97	0.96	0.97	79.51	99.86	1.00
8	698	2	4	99.53	R3	0.97	0.96	0.97	79.66	99.57	0.99
13	698	2	3	99.24	R1	0.66	0.64	0.66	66.76	98.14	0.70
13	698	2	3	99.24	R2	0.66	0.64	0.66	64.61	98.14	0.86
13	698	2	3	99.24	R3	0.66	0.66	0.66	65.33	99.43	0.94
20	698	2	5	99.61	R1	0.45	0.44	0.45	80.23	99.28	0.91
20	698	2	5	99.61	R2	0.45	0.44	0.45	78.22	99.57	0.57
20	698	2	5	99.61	R3	0.45	0.43	0.45	78.37	99.71	0.00[a]

Note: The table shows the simulation results per item and rater, the relative coverage of the tokens in the semantic space, the interrater reliability between the human raters κ_{HH}, between (all) simulated supported raters κ_{SS}, and between a single supported rater and the other human rater κ_{HS}. Additionally, the table indicates the proportion of automatically coded responses, the proportion of effort reduction *ER*, the proportion of correctly coded responses (General Accuracy; including manual and automatic codings), and the agreement between the automatically predicted scores with the true scores κ_{AC}.
[a]Due to uniform automatic coding, the kappa value cannot objectively reflect the agreement, since kappa results in 0, even with high accuracy.

TABLE A5 ASAP simulation results

Item	n	Codes	Median tokens	Cove-rage in %	Rater	κ_{HH}	κ_{HS}	κ_{SS}	ER in %	General Acc. In %	κ_{AC}
1	1672	4	45	99.27	R1	0.86	0.81	0.86	7.95	95.81	0.30
1	1672	4	45	99.27	R2	0.86	0.81	0.86	7.95	95.81	0.30
2	1278	4	59	99.09	R1	0.80	0.8	0.80	0.00	100.00	–
2	1278	4	59	99.09	R2	0.80	0.8	0.80	0.00	100.00	–
3	1891	3	49	99.39	R1	0.60	0.58	0.60	3.23	98.63	0.20
3	1891	3	49	99.39	R2	0.60	0.55	0.60	10.42	95.45	0.00[a]
4	1738	3	40	98.97	R1	0.61	0.55	0.61	23.76	91.94	0.39
4	1738	3	40	98.97	R2	0.61	0.55	0.61	20.66	92.87	0.22
5	1795	4	22	96.41	R1	0.91	0.69	0.91	49.3	92.09	0.04
5	1795	4	22	96.41	R2	0.91	0.75	0.91	45.35	94.43	0.00[a]
6	1797	4	19	96.34	R1	0.89	0.80	0.89	53.09	97.50	0.00[a]
6	1797	4	19	96.34	R2	0.89	0.83	0.89	51.47	97.66	0.00[a]
7	1799	3	37	99.17	R1	0.93	0.88	0.93	11.67	96.50	0.00[a]
7	1799	3	37	99.17	R2	0.93	0.88	0.93	11.67	96.33	0.00[a]
8	1799	3	48	98.88	R1	0.75	0.73	0.75	6.95	97.50	0.51
8	1799	3	48	98.88	R2	0.75	0.73	0.75	4.50	98.39	0.37
9	1798	3	41	98.50	R1	0.71	0.65	0.71	17.91	93.21	0.39
9	1798	3	41	98.50	R2	0.71	0.67	0.71	12.96	95.27	0.33
10	1640	3	33	98.28	R1	0.81	0.78	0.81	27.44	94.39	0.65
10	1640	3	33	98.28	R2	0.81	0.77	0.81	26.40	94.33	0.59

Note: The table shows the simulation results per item and rater, the relative coverage of the tokens in the semantic space, the interrater reliability between the human raters κ_{HH}, between (all) simulated supported raters κ_{SS}, and between a single supported rater and the other human raters κ_{HS}. Additionally, the table indicates the proportion of automatically coded responses, the proportion of effort reduction *ER*, the proportion of correctly coded responses (general accuracy; including manual and automatic codings), and the agreement between the automatically predicted scores with the true scores κ_{AC}.
[a]Due to uniform automatic coding, the kappa value cannot objectively reflect the agreement, since kappa results in 0, even with high accuracy.