Using and Improving Coding Guides For and By Automatic Coding of PISA Short Text Responses

Fabian Zehner[1,3]
[1]Technische Universität München
TUM School of Education
Arcisstr. 21, 80333 Munich, Germany
Email: fabian.zehner@tum.de

Frank Goldhammer[2,3]
[2]German Institute for
International Educational Research
Frankfurt am Main, Germany

Christine Sälzer[1,3]
[3]Centre for International
Student Assessment (ZIB) e.V.
Munich, Frankfurt am Main,
Kiel, Germany

Abstract—**We propose and empirically evaluate a theoretical framework of how to use coding guides for automatic coding (scoring) and how, in turn, automatic coding can enhance the use of coding guides. We adopted a recently described baseline approach to automatically classify responses. Well-established coding guides from PISA, comprising reference responses, and its German sample from 2012 were used for evaluation. Ten items with 41,990 responses at total were analyzed. Results showed that (1) responses close to the cluster centroid constitute prototypes, (2) automatic coding can improve coding guides, while (3) the proposed procedure leads to unreliable accuracy for small numbers of clusters but promising agreement to human coding for higher numbers. Further analyses are still to be done to find the optimal balance of the implied coding effort and model accuracy.**

Index Terms—**Automatic Coding, Automatic Scoring, Clustering, Coding Guides, Reference Texts**

I. Introduction

Short text responses play a crucial part in educational assessment. Having evolved from the field of automatic essay scoring [1], technologies for automatically evaluating short text responses have made vital progress in the last two decades. In this study, we adapt a recently described baseline approach [2] by reducing human involvement for model training. To satisfy requirements of machine learning procedures, most systems rely on relatively large amounts of manually coded training sets (often referred to as annotated data). These are not only expensive but might also contain incorrect codes, mainly due to the required mass. Hence recently, different research groups have strived to find appropriate procedures to train models with less but most informative data (cf. [3], [4], [5], [6]).

The reader will notice we use the psychometric term *coding* instead of the common *scoring*, *grading*, or *marking*. To us, the latter are a special case of coding which means to assign an entity to a category, whereas scoring means to additionally order these categories. The scope of automatic systems should not be limited to scoring only, although it is an important field of application. Analogously, with the term *coding guides* we refer to documents often used in social science studies that specify which class a response should be assigned to.

We argue that, at least, established assessments such as the *Programme for International Student Assessment* (PISA) already use documents for their manual coding that can be used to start training automatic systems. The so-called *coding guides* comprise item-specific *reference responses*: exemplary responses that are intended to constitute prototypes for their respective codes (i.e., full, partial, or no credit). Vice versa, experience teaches that empirical data will always force coding guide writers to update the coding guides during coding by adding new reference responses for two reasons—first, when a response type had not been considered and, second, to clarify the border between similar responses with different codes. Both can be supported by automatic systems identifying response types in the empirical data. This way, the coding guides help the automatic system to become trained, and the automatic system helps the coding guides to improve. Newly added reference responses then only need to be assigned to the intended code by the coding guide writers according to the assessed construct. Essentially, the number of new reference responses needs to be manageable, enabled by proper automatic identification of response types. Thus, the human is used for what the human is really good at—namely assigning a few single responses to codes—and the computer is used for what the computer is really good at—namely first sampling a few informative responses out of a mass and later on applying the learned rules to a mass. This procedure can also be used to simply systematically create coding guides from scratch.

The present study demonstrates (1) the use of coding guides as a source of training data compared to training with completely manually coded data and (2) how automatic coding via clustering improves coding guides. It also (3) examines the use of cluster centroids, in that, we show (a) how to sample prototypical responses and (b) that cluster centroids constitute representative prototypes for this. Finally (4), it shows the accuracy development in relation to the number of clusters. In this paper, accuracy is operationalized as human–computer agreement.

II. Procedures for Automatic Coding

This section first describes the basic approach taken to automatic coding and then, second, proposes adaptions. Third, the problem of sampling prototypes is elaborated. Finally, the employed system is compared to existing ones.

A. Semantic Clustering as Automatic Coding

The automatic coding system described in [2] was used. Briefly outlined, it first builds a vector space model using Latent Semantic Analysis [7] on the basis of a text corpus that is especially sampled for the respective item semantics (resulting in one corpus per item). Next, the empirical responses are preprocessed using common techniques such as stemming and spelling correction. The *bag of words*–paradigm is applied. Each response is then represented by the semantic centroid vector of all its tokens in the semantic space. In a next step, these response vectors are clustered by a hierarchical, agglomerative cluster analysis. The required number of extracted clusters can either be determined on the basis of manually assigned codes using stratified, repeated tenfold cross-validation (supervised) or with regard to the development of the clustering rest criterion (unsupervised). Once built, this cluster model serves to automatically code unseen responses by finding the most similar cluster centroid and assigning the cluster code. The cluster code is computed on the basis of manually assigned codes. This last step is where our proposed adaptation comes in.

B. The Use of Coding Guides for Automatic Coding

Requiring completely manually coded data to determine the cluster codes bears the disadvantages already described. Therefore, we propose the following to minimize the manual coding effort. First, the unsupervised variant is chosen to determine the appropriate number of clusters. These represent different response types. Second, the reference responses from the coding guides are processed in the same way as the empirical responses. Third, the reference responses are projected into the semantic space and each is assigned to the most similar cluster. Ideally, all clusters now have unambiguously coded reference responses assigned. In such a case, the final model is attained and can serve for automatic coding. But in most real-world cases, at least some of the conflicts described below are likely to appear. Once the conflicts have been solved, again a final model is attained offering the possibility for automatic coding.

The following conflicts are worth examining. They might either indicate difficulties for proper automatic coding or insufficient coding guides. For some item types, the approach to automatic coding presented here is simply not appropriate. But in principle, both can be improved by looking at two diagnostics: the frequency distribution of reference responses across clusters and the distribution of response distances to their cluster centroid within clusters. With regard to the former, the perfect coding guide would assign one reference response to one empirically evolving type. But for clusters lacking a reference response (Conflict I), a new reference response needs to be sampled from the empirical responses. Next, this new reference response is manually coded by the coding guide writers.

In other cases, the reference responses from coding guides concentrate on a few or even a single response type and, hence, a single cluster. This is not ideal but not necessarily a problem. Such a case only presents a conflict if these reference responses are intended to capture different response types. This might reveal an inappropriate semantic space and should raise awareness. Even worse, reference responses assigned to the same cluster could belong to different codes (Conflict II). This would reveal an insufficient semantic space or cluster model.

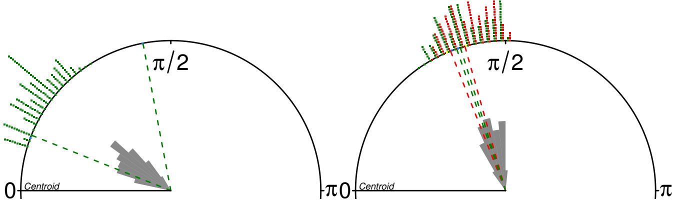

Figure 1. Exemplary Rose Diagrams; visualization of one accurate and one inaccurate item #6 cluster. The horizontal lines constitute the cluster centroids and the points represent responses within the clusters and their distance to the centroid (clockwise). Green points are manually coded as full credit, red ones as no credit. The dashed lines depict coding guide reference responses.

Moreover, the cluster-wise distributions of all response distances to their cluster centroid are informative diagnostics. Distances between two vectors x and y are operationalized as arccosine, $\Delta_{\vec{x},\vec{y}} = arccos(\frac{\vec{x}\cdot\vec{y}}{|\vec{x}|*|\vec{y}|})$, and, hence, the within-cluster distance of a response $\vec{r}$ assigned to cluster i with centroid $\vec{c_i}$ is given by $\Delta_{\vec{c_i},\vec{r}}$. Half rose diagrams [8] can help to visualize the distribution and the reference response's relation to it (cf. Figure 1). These diagrams are histograms for circular data with the adaptation that the coefficient's range is $[0, \pi]$, thus, only a semicircle. When considering a reference response's position within this distribution, it is considered a prototype for this cluster if it is close to the centroid, indicated by a relatively small distance; this would show a good fit between the reference and the empirical responses. Other reference responses might represent response types not occurring in the empirical data at all if they are at the right tail's end of the distance distribution (Conflict III). These clusters then need a new reference response analogous to Conflict I if they have no prototypical reference responses assigned. The next subsection describes this sampling process.

C. Sampling Prototypes

One crucial step in the new procedure is to determine which empirical responses should be sampled as additional reference responses. When a regression is carried out on the sampled responses, optimal design algorithms, such as the Fedorov exchange algorithm [9], are highly effective [4]. That is because they select the most informative responses for the regression, those at the distribution's periphery. In the case of an unsupervised clustering has already been carried out, the researcher does not need to know analogously about the informative border responses. Instead, the researcher is in the fortunate situation of knowing about the data's underlying structure and, hence, knows about response types—basically, any response from each cluster could be sampled. Unfortunately, the number of clusters is not deterministic but ambiguous and is desired to be relatively small. This often results in clusters comprising more than one response type in a strict sense. For clustering approaches, the overall goal is to identify one response that is most prototypical for the whole cluster. Often, responses close to their centroid are simply assumed to be the most prototypical (cf. [2], [3]). In [5], a list heuristic is used to find the response with the highest similarity *and* the most connections. In other terms, the heuristic seeks for the densest area within a cluster. This appears to be the most evidence-based reasoning. In the present study, we show empirical evidence of responses that should be sampled as prototypes for their cluster.

When striving to find a cluster's densest area, the most elegant way might be kernel density estimates. However, in most applications of automatic coding, these cannot be used because too many dimensions are employed resulting in a space that is too sparse. To examine dense areas in clusters, we here adopt an approximation similar to the list heuristic used in [5], making use of the fact that dense areas comprise many responses with relatively low pairwise distances. This is analogous to the definition of dense areas in kernel density estimation searching for the smallest area with the highest density (cf. [10]). Thus, we sort responses' pairwise distances to other responses within the cluster increasingly. These are then plotted per response. The responses with the most relatively low distances belong to the densest area. This approach guarantees to find local dense areas, but it does not guarantee that these responses do not just constitute a dense area within the cluster's periphery.

D. Related Work

The basic approach of the automatic coding employed in this study is to cluster vectors that represent the semantics of the responses [2]. The underlying concept is that the response type is the result of the respondent's cognitive processing and reaction to the question. Both the heterogeneity of respondents and the characteristics of the question determine evolving response types. Hence, responses can be grouped into question-specific types and most questions allow more than one type of correct responses (i.e., different lines of reasoning or different wordings).

A comparable system called *Powergrading* has recently been developed and partly attained remarkable performance [11]. It applies two-level clustering and learns a distance metric by a supervised method. Despite the powerful methodological approach taken, one reason for the high accuracy might be that the analyzed questions evoked a similarly low language diversity in the responses as one of the items analyzed in [2], which was also automatically coded excellently. *Powergrading* uses a fixed number of clusters, which is not plausible for typical assessment needs where different questions naturally tend to evoke different numbers of response types.

The unsupervised system described in [12] also makes use of similarities between responses based on Latent Semantic Analysis but does not group them by clustering. The authors propose to enrich the original response key with the words from the empirical responses that have the highest similarity to it. Whereas this procedure is comparable to our mechanism which adds empirical responses as new reference responses to the coding guide, it conceptually appears not to be optimal to us. Instead of allowing for different lines of reasoning, this might only add synonyms to the response key which already should have been considered similar by the vector space model.

A relatively large body of works dealing with the automatic grading of assignments recently has been evolving around massive open online courses (MOOCs). With some having tens of thousands of student assignments, they are a natural and important field of application for automatic coding. Which automatic systems are applicable highly depends on the kind of responses that are analyzed. For example, some systems were developed to automatically grade programming source texts. Some of these systems take comparable approaches to ours as similarities between students' and key responses are in the center of interest (e.g., [13], [14]). Some also apply clustering methods on these similarities (e.g., [15]). But the essence of how similarity is operationalized differs crucially. These systems need to deal with characteristics of formal language. When responses predominantly are natural language, as is the case for our needs, the systems mainly require information about the semantics of words (such as in [2], [11], [12], [16]).

Since the processing of natural language is relevant to various different domains—for example, dialog systems—a vast diversity of implementations with different adaptations is available in literature. For instance, one interesting concept of weighting single words in vectors of a vector space model are extrema, proposed in [17]. Yet, the authors found that extrema do not outperform the baseline in the domain of questions.

Furthermore, one important objective of our research is to implement automatic coding in the assessment area of social sciences. Thus, we decided to design our system in a way that is accessible and transparent in terms of a clear understanding of the underlying procedures. As the focus of social sciences is mainly content-related, it is worthwhile to be able to incorporate the outputs into the research's theory, such as response types represented by clusters. This is often hardly feasible when applying more powerful machine learning methods such as support vector machines or neural networks where the outputs are, besides the desired classifier, feature weights and neuron thresholds that are difficult to interpret. Hence, we selected clustering and Latent Semantic Analysis, which is conceptually related to factor analysis, since both these techniques are close to common methods in social sciences. Moreover, utilized unsupervised clustering is the most natural method of applying our main concept described above—questions evoke different response types that correlate with the respondent's cognitive reaction to the question.

III. Methods

This section briefly describes the materials used and data collected as well as the employment of the system and the analyses carried out.

Table I
ITEM CHARACTERISTICS

Item	Domain[a]	Aspect[b]	Correct	n	Words[c]
#1	read	B	83%	4,152	12.3 (4.6)
#2	read	C	43%	4,234	15.6 (9.0)
#3	read	B	10%	4,234	12.5 (6.3)
#4	read	A	59%	4,223	5.6 (3.0)
#5	read	C	56%	4,234	14.7 (6.2)
#6	read	B	80%	4,152	12.4 (6.9)
#7	read	C	68%	4,152	13.6 (7.0)
#8	read	B	69%	4,223	14.4 (5.5)
#9	math	M	35%	4,205	14.0 (6.8)
#10	scie	S	58%	4,181	11.1 (5.2)
Total			56%	41,990	12.6 (6.1)

[a] one of *read*ing, *math*ematics, *scie*nce
[b] A = Access & Retrieve, B = Integrate & Interpret, C = Reflect & Evaluate, M = Uncertainty & Data, S = Explain Phenomena Scientifically (according to PISA framework [20])
[c] average word count in nonempty responses (SD)

Table II
EXEMPLARY PISA ITEMS

Question	Full Credit Response	No Credit Response
One part of the article says, “A good sports shoe should meet four criteria.” What are these criteria?	It must provide exterior protection, support the foot, provide the player with good stability and must absorb shocks.	Protect against knocks from the ball or feet. 2. Cope with unevenness in the ground. 3. Keep the foot warm and dry. 4. Support the foot.
Why does the article mention the death of Kiyoteru Okouchi?	To give the background to why people are so concerned about bullying in Japan.	It’s just to grab your attention.

Note. Further released items and details such as the stimulus texts can be found in [21] (cf. pages 53 and 60 for the two given items).

A. Materials and Data

The data and coding guides used in the presented analyses stem from the German PISA 2012 sample. This includes a representative sample of 15-year-old students as well as a representative sample of ninth-graders in Germany. A detailed sample description can be found in [18] and [19]. Due to a booklet design, the numbers of test takers varied for each item ($4152 \geq n \geq 4234$). In PISA 2012, reading, maths, and science were assessed paper-based. That is why not all items but only ten transcribed ones, including eight reading, one maths, and one science item, were at hand. All items were coded dichotomously, that is, responses either got full or no credit. Table I presents some more item characteristics. More details on the assessed constructs and the transcription procedure can be found in [2]. Item and response contents cannot be reported due to the items’ confidentiality. An example of two typical PISA items and respective responses are given in Table II; these were not part of the analysis. With regard to the kind of responses that are evoked, the first example is representative for item #4 and the second one for all others.

B. Employment of the Theoretical Framework

In the analysis presented here, we applied the described theoretical framework as follows. In order to decide on a cluster code, the system takes into account the coding guide reference responses assigned to this cluster. Then, those responses with a distance to the cluster centroid of at least the cluster-specific distribution’s mean plus 1.6 standard deviations are omitted due to them being insufficiently prototypical. This way, the reference responses farther away from the centroid than 95 percent of all responses in the cluster are not used as they do not have empirical equivalents. If the remaining prototypical reference responses all have the same code, this code is used as the cluster code. If there are contradicting codes, however, the cluster is flagged for manual inspection and the code with the highest frequency within these prototypes is used as the cluster code. In case of a tie between codes (i.e., none of the codes reaches a majority), the reference responses are not used at all but a new empirical reference response has to be sampled as a prototype. This is also true for cases in which no reference response is assigned to the cluster at all.

In cases in which a new prototype had to be sampled, we selected the k responses closest to the respective cluster centroid. To analyze how the procedure works out with minimal coding effort, we set k to 1. In case the resulting codes differ, the semantic space needs to be analyzed manually. As the data we used in the analysis already had been completely manually coded by humans, we did not need to code the sampled responses but just used the manual code.

C. Analyses

All analyses used the default parameter setup suggested in [2] including stemming, spelling correction, 300 dimensions in the vector space model, cosine as the distance metric, and Ward’s method for agglomeration. Analysis I investigates the required changes for the coding guides to cover all the empirical response types. The conflicts as described above that arose are depicted. In Analysis II, we show empirical evidence how prototypes should be sampled from clusters in case no reference response is available.

Analysis III follows two interests. First, it studies the system’s accuracy when trained by the proposed procedure using coding guides (`cg`) compared to the accuracy when trained by the completely manually coded data (`man`). Cohen’s kappa ($\kappa_{h:c}$) and the coefficient $\lambda_{h:c}$ introduced in [2] are reported for each condition. The latter is a corrected coefficient of percentage of agreement giving the proportion of the actual human–computer agreement’s increase to the highest attainable increase: $\lambda_{h:c} = \frac{\%_{h:c_i} - \%_{h:c_1}}{100 - \%_{h:c_1}}$. For this, the percentage of agreement for the model with only one cluster ($\%_{h:c_1}$) is subtracted from the percentage of agreement for the final model with i clusters ($\%_{h:c_i}$). This difference is then divided by 100 minus the percentage of agreement for the model with one cluster, constituting the highest attainable increase. Taking a conservative approach, we overestimated the accuracy in the `man`-condition because all data were used for training and testing simultaneously and, hence, constitute a difficult benchmark to reach. This was necessary because the `cg`-

Table III
CODING GUIDE CONFLICTS

	Number of		Conflict		
Item	**Clusters**	**Ref. Resp.**	**I**	**II**	**III**
#1	52	17	39 (75%)	1	7 (41%)
#2	48	21	34 (71%)	1	5 (24%)
#3	70	31	50 (71%)	1	7 (23%)
#4	7	17	1 (14%)	2	3 (18%)
#5	53	32	32 (60%)	2	7 (22%)
#6	53	21	39 (74%)	0	4 (19%)
#7	46	15	35 (76%)	0	3 (20%)
#8	31	17	22 (71%)	1	4 (24%)
#9	46	12	37 (80%)	1	1 (8%)
#10	55	15	44 (80%)	2	1 (7%)
Total	461	198	333 (72%)	11	42 (21%)

Note. Conflict I: clusters without coding guide reference response (percentage relative to number of clusters), II: cases in which reference responses with contradicting codes were assigned to the same cluster, III: reference responses without empirical correspondence (percentage relative to number of reference responses)

condition used the whole data for training and applying a cross-validation here might have introduced artificial effects.

Second, Analysis III examines the accuracy in relation to the number of clusters, comparable to the learning curve analysis in [3]. The number of clusters directly implies the coding effort. Particularly in this second part of the analysis, $\lambda_{h:c}$ is the optimal measure because it primarily indicates accuracy increase being corrected for a stable overall agreement by chance as well as empty responses and, thus, sensitively shows up accuracy changes. In comparison, $\kappa_{h:c}$ is an unstable measure in this context as its range depends crucially on marginal totals, which vary for each run. Therefore, it should be interpreted with awareness; nevertheless, we additionally report on this scale as most readers will be familiar with it.

IV. RESULTS

This section is structured by the different analyses described in the previous section.

A. Analysis I: Improvement of Coding Guides by Automatic Coding

The numbers of clusters were chosen with regard to the development of the rest-criterion. Table III depicts the number of clusters by item, the number of reference responses extracted from the coding guides, and how many conflicts occurred. In rare cases (II: 11 of 198), reference responses with different codes were assigned to the same cluster indicating an insufficient model. Often, this was due to generally improper automatic coding within the items; that is, when the system neglects a relevant linguistic information impacting a response's code, such as negation might. Another reason for these conflicts are the relatively small numbers of clusters as forcing clusters to join might result in the mixing of reference responses with different codes.

Partially, Conflict III cases can similarly stem from imperfect automatic coding; indeed, the three items #1, #3, and #5 performing poorest according to [2] show up with 22–41 percent of reference responses that are discarded because they are very remote from their cluster centroids. But generally across items, with the exception of the math and science item, there appears a relatively high tendency of 21 percent of reference responses that cannot be mapped to empirical response types. For an exemplary visualization, refer back to the left part of Figure 1 where there is one very prototypical reference response, represented by the dashed line on the left. Also, there is a reference response assigned to this cluster that is way apart from the cluster's distribution. Furthermore, in this figure, the empirical grounding for the procedure described in Section II-B, to omit reference responses if they are not prototypical, can be found. Although one would assume the distribution to be half of a very steep normal distribution around the centroid, the distributions very much behave like normal distributions with their mean shifted at the range of $[\frac{\pi}{4}, \frac{\pi}{2}]$. This is, amongst others, due to the vectors' hyperdimensionality and is additionally dependent on the number of responses in the cluster. Moreover, the distance distribution can be used, for example, as an indicator for the cluster's homogeneity or often even purity—obviously the distribution of the inaccurate cluster (mixing codes; cf. right part of Figure 1) is more shifted away from the centroid than the pure cluster (cf. left part of the figure).

High rates of clusters do not have reference responses assigned to them at all (I: 72% on average). Considering the discrepancy between the number of reference responses available in the coding guides and numbers of clusters extracted, this might not be surprising. Also, the automatic coding generally distinguishes more response types than humans do because its approximation of language comprehension is highly superficial. Nevertheless, different clusters exist due to concrete differences on the language level, which might influence human coders, so the values of up to 80 percent show a high potential for coding guide improvements.

B. Analysis II: Sampling Prototypes

For all items and clusters, the procedure described in Section II-C was conducted. Patterns of pairwise distances of responses were analyzed as shown in Figure 2 to identify dense regions within clusters. Two exemplary clusters are given, a bigger and a smaller one. Each line represents one response within one cluster of a specific item. As the increasingly ordered distances are plotted, those curves of responses correspond to dense areas that are relatively low as long as possible with regard to the x-axis. Such responses have many low distances to other responses and, thus, are member of a relatively dense area within the cluster. Additionally, the five responses that are closest to the cluster centroid stand out in black, dashed lines.

Obviously, the responses close to the cluster centroid are located in the relatively densest areas. This finding is exceptionally consistent across all items and clusters. The list heuristic, on the other hand, ensures to find the densest areas but not necessarily one that is prototypical for all other responses in the cluster. It is conceivable to find a small dense area at the cluster's periphery that is not representative for the rest of the

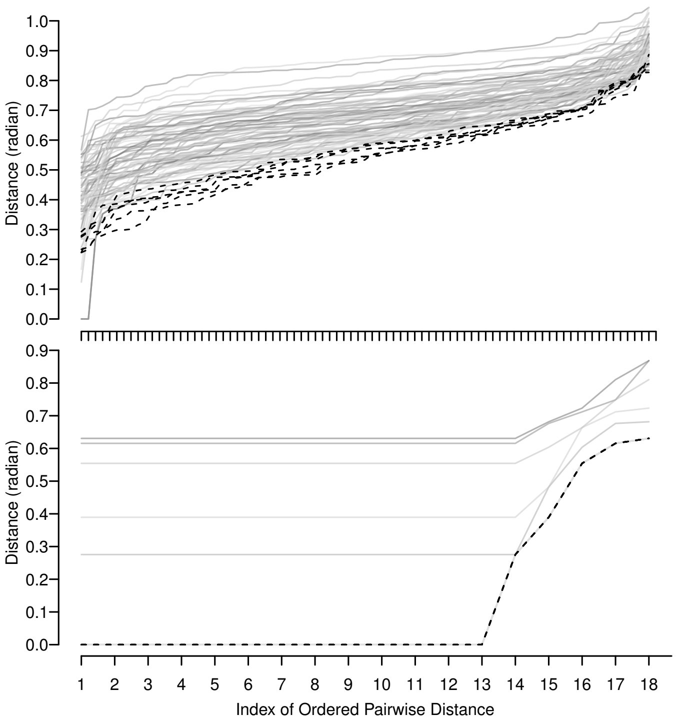

Figure 2. Identifying Prototypes. Two exemplary figures of increasingly ordered pairwise distances of responses within one cluster (item #10, clusters 3 [top] and 19 [bottom]). Each line constitutes one response, the black, dashed ones are the five responses closest to the cluster centroid.

cluster. Hence, in rare cases, such a list heuristic as used here and in [5] can be misleading; an example can be found at the upper part of Figure 2 where the lowest lines are two similar responses of which the following distances are relatively high. Therefore, we recommend to use the responses close to the centroid for sampling of prototypes. These constitute the optimal prototypes as they are always in dense areas and at the same time at the cluster's center guaranteeing to be the best representatives of all cluster members.

C. Analysis III: Comparison of cg and man and Learning Curve

The first part of this analysis focuses on the accuracy of the cg-condition compared to the man-condition. The results can be found in Figure 3 by comparing the top row (man) with the bottom row (cg). Basically, the system's accuracy drops remarkably by using the proposed approach, condition cg. The accuracy seems to randomly jump instead of steadily improving along increasing cluster numbers. We assume that our goal to minimize the simulated coding effort, using $k = 1$ for the prototype sampling, gave too much weight to single response codes. These could influence many other responses in cases of large clusters. This is supported by Figure 3 showing the curves' stabilization at $\geq$100 clusters. These larger numbers of clusters result in smaller clusters, and thus, in a reduced impact of single responses on other responses. The accuracy values of the two conditions in these higher ranges of numbers of clusters do not differ greatly but more analyses with more reliable, and thus more comparable, setups are necessary.

In the second part of the analysis, we concentrated on the relationship between accuracy and number of clusters. Again, the results can be found in Figure 3. Contrary to our intention, the results of the man-condition should be interpreted in the first place because the cg-condition's results seem partially imprecise. Nevertheless, the findings can also be mapped to the cg-condition—if we assume the worst case scenario, where no reference response was accessible to automatically code the empirically found response types, each cluster that is extracted entails more coding effort. Yet, every bit of information helps to build more reliable models. This trade-off is obvious in the results. Generally, it can be seen that the more clusters are extracted the higher the agreement will be. This is not surprising, particularly as the test data is identical to the training data, as explained previously. Still, this is not the crucial aspect but it is obvious that a lot of different response types can be found in about 4,200 responses. Item #4 here serves as a good showcase. It represents the least complex item type in PISA in which the test taker only needs to repeat information that is explicitly given in the stimulus, resulting in a very low language diversity in responses (cf. the item's average word count in Table I as an approximation). In this case, the test takers are asked to repeat a list of four terms. Although the automatic coding of this item already reaches a very good agreement with 7 clusters, it continues to show marginal improvements in the range of 280 clusters. Of course, the response types represented by these clusters only occur occasionally—the first cluster still carries 57 percent of nonempty responses. Yet, this case shows the language diversity that automatic coding needs to deal with in even such a simple setting. This underlines the importance of vector space models and their semantic concepts opposed to pure word-based processing.

Also interesting in Figure 3 is the steepness of the curves. A steep learning curve means that with few clusters a high gain in accuracy is attained. This can particularly be found for the items #4 and #2, which converge towards their optimum almost in the range of 10–20 clusters. Others show steady improvement, reaching their optima only at about 70–90 clusters. Still, there are items that are not properly coded and show a very low improvement.

V. Conclusion and Directions

The presented theoretical framework and empirical evaluation show that the established use of coding guides in assessments and the nascent field of automatic coding can benefit from each other. The automatic coding approach we took up from [2] is based on identifying response types that can be used as reference responses in coding guides. The first analysis showed a high potential of real coding guides employed in PISA to be replenished with further reference responses. These can be identified very efficiently by the operated automatic coding system without any manual coding

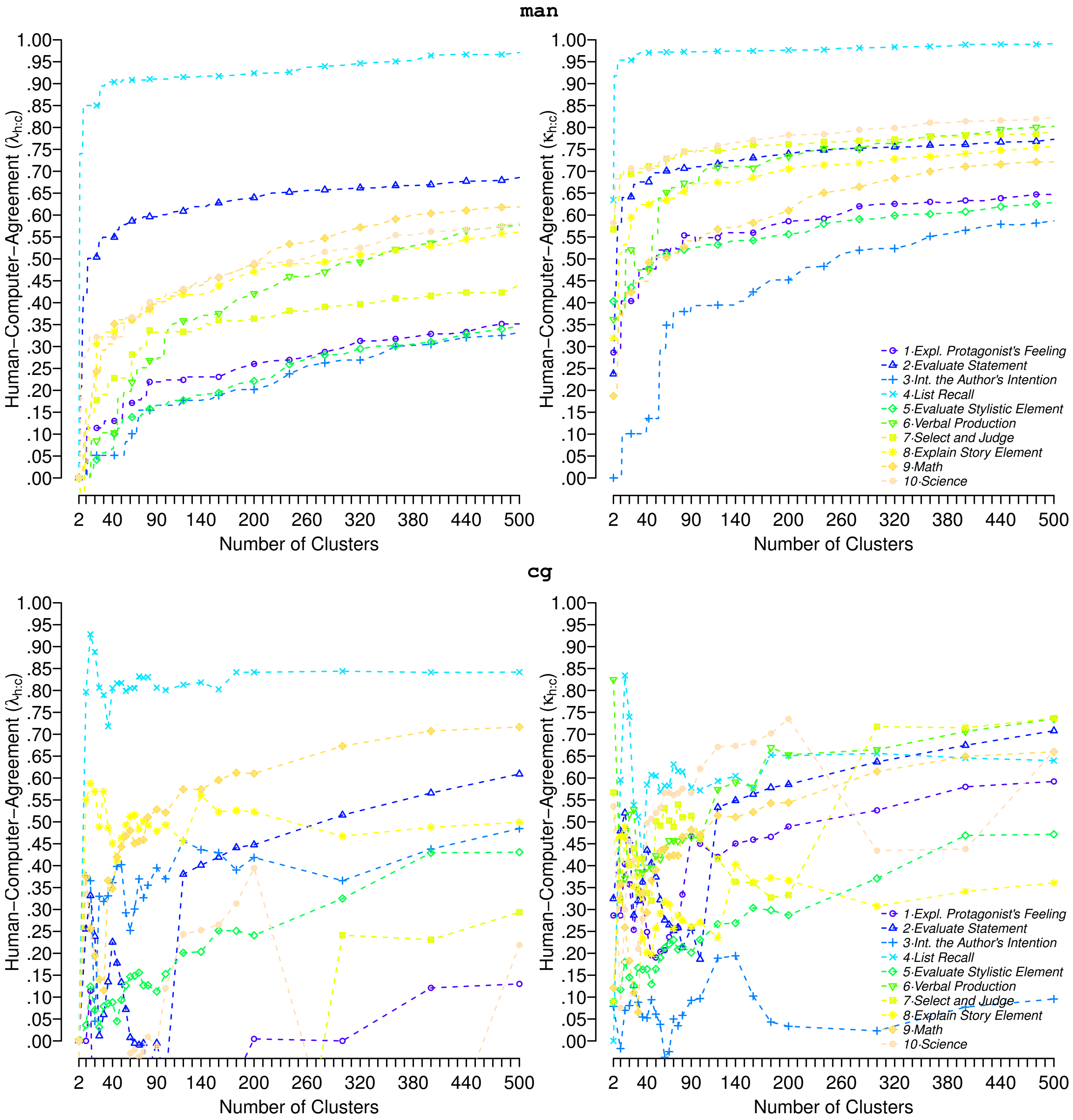

Figure 3. Accuracy by Number of Clusters and Condition. Human–computer agreement ($\lambda_{h:c}$ left, $\kappa_{h:c}$ right) for the `man`- (top) and `cg`-condition (bottom) plotted against the number of clusters being extracted for each item. The $\lambda_{h:c}$-coefficient indicates the relative accuracy increase compared to a solution with one cluster.

by humans, which not only costs time and money but is also prone to errors. The system puts only a baseline approach into practice leading to this advantage of unsupervised methods over most other, supervised automatic coding systems which require at least a minimum of such manual coding because they utilize powerful machine learning procedures such as support vector machines.

Conversely, the concept of coding guides, which are produced by the items' experts, most often the item developers themselves, is very promising to the field of automatic coding. Strongly reducing the manual coding effort allows to let experts decide on the coding of responses, or rather response types. The resulting training data would contain fewer errors due to exhaustion, inconsistency, or even misconceptions of trained coders (an elaborate, concise overview of rater cognition typically influencing manual coding can be found in [22]).

Despite the promising approach, the empirical accuracy of the automatic coding system showed unreliable variation using coding guides and sampling of new reference responses in the range up to 100 clusters. From this point, the system's performance becomes more accurate and reliable with not too much deviation from the original system. This evidence suggests that using only one newly sampled response as the prototype and as the decision on the code for the whole cluster ($k = 1$) leaves too much impact to chance. This is true, although we showed evidence that the sampled responses close to the cluster centroid are indeed most prototypical for their clusters. The combination of using only $k = 1$ responses for sampling to simulate minimal coding effort with only a few clusters, which in turn led to relatively large clusters, is likely to have produced the inaccurate performance. Hence, we suggest setting k at higher values such as 3 or 5. A first analysis showed the expected improvements but a more systematic analysis is necessary regarding how to optimally balance k and the numbers of clusters without overtaxing the manageable amount of manual coding effort. Nevertheless, the proposed approach appears promising when taking into account the relatively small loss in accuracy as opposed to the system that was trained with over 4,000 response codings in the range of $\geq$ 100 clusters. Even if no reference responses are usable for 100 empirical clusters, the effort to manually code 100 responses, which constitutes about 2% of the 4,200 codings that were needed otherwise, seem manageable for coding guide writers.

References

[1] E. B. Page, "The imminence of grading essays by computer," *Phi Delta Kappan*, vol. 48, pp. 238–243, 1966.

[2] F. Zehner, C. Sälzer, and F. Goldhammer, "Automatic coding of short text responses via clustering in educational assessment," *Educational and Psychological Measurement*, 2015. [Online]. Available: http://epm.sagepub.com/content/early/2015/06/06/0013164415590022

[3] T. Zesch, M. Heilman, and A. Cahill, "Reducing annotation efforts in supervised short answer scoring," in *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, Association for Computational Linguistics, Ed., 2015, pp. 124–132.

[4] N. Dronen, P. W. Foltz, and K. Habermehl, "Effective sampling for large-scale automated writing evaluation systems," *arXiv preprint arXiv:1412.5659*, 2014.

[5] L. Ramachandran and P. Foltz, "Generating reference texts for short answer scoring using graph-based summarization," in *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, Association for Computational Linguistics, Ed., 2015, pp. 207–212.

[6] J. Z. Sukkarieh and S. Stoyanchev, "Automating Model Building in c-rater," in *Proceedings of the 2009 Workshop on Applied Textual Inference*, 2009, pp. 61–69.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, and T. K. Landauer, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[8] S. R. Jammalamadaka and A. Sengupta, *Topics in circular statistics*, ser. Series on multivariate analysis. River Edge, N.J: World Scientific, 2001, vol. v. 5.

[9] V. V. Fedorov, *Theory of optimal experiments*, ser. Probability and mathematical statistics. New York: Academic Press, 1972.

[10] D. W. Scott, *Multivariate density estimation: Theory, practice, and visualization*, ser. A Wiley-Interscience publication. New York, NY: Wiley, 1992.

[11] S. Basu, C. Jacobs, and L. Vanderwende, "Powergrading: A clustering approach to amplify human effort for short answer grading," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 391–402, 2013.

[12] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 2009, pp. 567–575.

[13] A. Nguyen, C. Piech, J. Huang, and L. Guibas, "Codewebs: scalable homework search for massive open online programming courses," in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 491–502.

[14] S. Srikant and V. Aggarwal, "A system to grade computer programming skills using machine learning," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1887–1896.

[15] E. L. Glassman, R. Singh, and R. C. Miller, "Feature engineering for clustering student solutions," in *Proceedings of the first ACM conference on Learning@ scale conference*, 2014, pp. 171–172.

[16] S. Jing, "Automatic Grading of Short Answers for MOOC via Semi-supervised Document Clustering," in *Proceedings of the 8th International Conference on Educational Data Mining*, O. C. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, and M. Desmarais, Eds., Madrid, 2015, pp. 554–555.

[17] G. Forgues, J. Pineau, J.-M. Larchevêque, and R. Tremblay, "Bootstrapping Dialog Systems with Word Embeddings," Montreal, 12.12.2014. [Online]. Available: http://www.cs.cmu.edu/~apparikh/nips2014ml-nlp/camera-ready/forgues_etal_mlnlp2014.pdf

[18] M. Prenzel, C. Sälzer, E. Klieme, and O. Köller, *PISA 2012: Fortschritte und Herausforderungen in Deutschland.* Münster: Waxmann, 2013.

[19] OECD, *PISA 2012 results: What students know and can do - Student performance in mathematics, reading and science*, volume 1, revised edition, february 2014 ed. Paris: OECD Publishing, 2014.

[20] ——, *PISA 2012 assessment and analytical framework: Mathematics, reading, science, problem solving and financial literacy.* OECD Publishing, 2013.

[21] ——, "PISA Released items - reading," 2006. [Online]. Available: http://www.oecd.org/pisa/38709396.pdf

[22] I. I. Bejar, "Rater cognition: Implications for validity," *Educational Measurement: Issues and Practice*, vol. 31, no. 3, pp. 2–9, 2012.