Towards Design-Loop Adaptivity: Identifying Items for Revision

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

Tomáš Effenberger
Masaryk University Brno
tomas.effenberger@mail.muni.cz

Adam Kukučka
Masaryk University Brno
524905@mail.muni.cz

We study the automatic identification of educational items worthy of content authors' attention. Based on the results of such analysis, content authors can revise and improve the content of learning environments. We provide an overview of item properties relevant to this task, including difficulty and complexity measures, item discrimination, and various forms of content representation. We analyze the potential usefulness of these properties using both simulation and analysis of real data from a large-scale learning environment. We also describe two case studies where we practically apply the identification of attention-worthy items. Based on the analysis and case studies, we provide recommendations for practice and impulses for further research.

Keywords: learning environment, outliers, anomaly detection, interpretability, reliability, difficulty, content analysis, attention-worthiness

1. Introduction

Practically used learning environments contain extensive content, particularly when they use personalization features (personalized learning requires significantly more content than the common textbook approach). The management of a large content base is complex and requires iterative improvement (Pelánek, 2020a). To realize this improvement, it is advantageous to leverage the complementary strengths of human intelligence and machine learning techniques based on data analysis. Such techniques are sometimes denoted using the terms *human-in-the-loop*, *intelligence augmentation*, or *design-loop adaptivity* (Aleven et al., 2016; Baker, 2016; Hassani et al., 2020; Doroudi, 2019).

Previous research work in this direction consists mainly of closing-the-loop studies (Liu and Koedinger, 2017), where analytics results are used to improve the environment (by a human) and the new version is experimentally validated. These studies provide valuable research-based insights, but they are hard to perform in practice as they require expertise. Thus, their practical impact has been limited so far. Our aim is to explore intelligence augmentation techniques that are applicable in the development of practically used learning environments that contain tens of thousands of items organized into hundreds of topics.

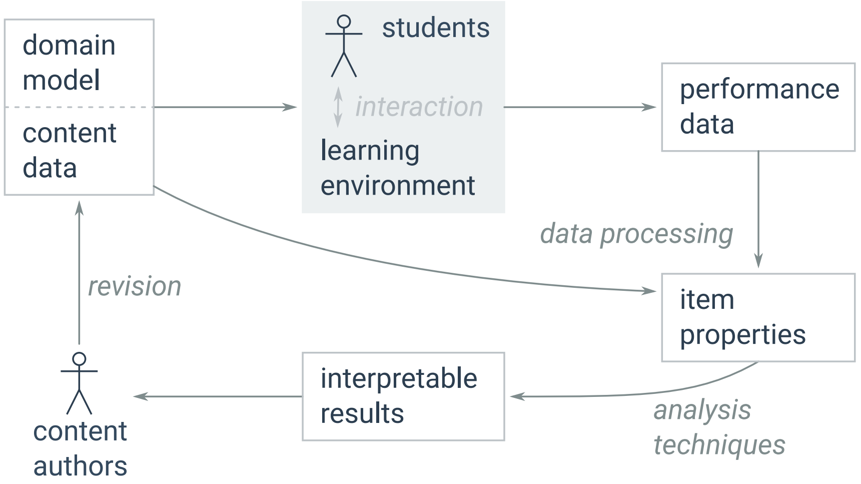


Figure 1: Illustration of the context and flow of data.

To clarify the high-level context of this work, Figure 1 illustrates the specific design-loop adaptivity approach that we consider. We assume an interactive learning environment that uses content data (e.g., questions, problems, explanations) and domain data (e.g., mapping of questions to topics) and collects data on student performance. Based on these data, we want to obtain interpretable results about the content, which will help content authors revise and improve the available content.

The specific goal that we address is the automatic identification of items that are *worthy of the attention* of content authors. There may be many reasons why an item may be worthy of attention. Typically, such an item is in some way "bad": it may be wrongly specified, contain a mistake, have a poor or misleading formulation, or it may not fit into the specific context in which it is used. But it may also be a good item that illustrates under-practiced content.

Provided with an attention-worthy item, content authors may choose to perform one of several actions, e.g., remove the item, revise the formulation, add an explanation, add other similar items, or create a new topic. To facilitate the decision on the appropriate action, the detected items should ideally be accompanied by some interpretable analysis that explains why the item is worthy of attention. Once experts are shown difficulty statistics for items, they may see problems that they would miss without these statistics (see, for example, Wang et al. (2021)).

To illustrate our aim, Table 1 provides specific examples of the type of attention-worthy items that we want to detect. All of them are real-life examples from a learning environment that we use for experiments in this work and were detected using some versions of the below-described automated techniques. The reasons for attention-worthiness and possible actions were provided by content authors. Note that the reasons for attention-worthiness depend on the specific context in which the items were used (e.g., other items used in the specific item set), which cannot be fully reproduced in detail here; we can provide only a brief summary.

This work brings several contributions. We explicitly formulate the problem of "detecting items worthy of attention," which we believe is an interesting direction for educational data mining as it combines interesting research challenges and important practical value. We propose a set of item properties that can be used to tackle this problem, organizing them into properties based on content and properties based on student performance. We describe several exploratory

Table 1: Examples of specific items worthy of attention with a description of possible actions that can be taken by content authors.

item and topic	**why worthy of attention**	**possible action**
Can `not x != y` be simplified to `x = y`? *Logic expressions in Python*	Even high-performing students answer incorrectly. The question is too tricky.	Remove the item.
Is `2nd_attempt` a valid name of a variable? *Variables and expressions in Python*	High error rate and low response time. Other items focus on semantics, this one concerns syntax.	Either remove the item or add more items of the same type.
My cat [is/are] grumpy. *To do, to have, to be in present simple*	The item is focused on elementary grammar, but uses advanced vocabulary ("grumpy").	Revise the item (use another adjective).
I [don't understand / am not understanding] what you are saying. *Present simple vs. present continuous*	The sentence combines actions of two persons; most other sentences mention only one actor.	Add more items of this type.
$(\sqrt{2} \cdot 2)^2$ *Expressions with powers and square roots*	The item is difficult; most other examples in the set contain either power or square root, but not both.	Add more items of this type.
$8 - (8 + t)$ *Simplifying expressions*	The item is difficult and most students make the same mistake.	Add explanation for the common mistake and more items of the same type.
$3.6 - 4$ *Addition and subtraction of decimal numbers*	This is the only example in the used set of items where the answer is negative number; this probably confuses students.	Remove item from the current set, add a new topic with negative numbers.

Are the following expressions equivalent?
`x > y-5 and x < y+5`
`abs(x-y) < 5`
Logic expressions in Python
	The item has high response time. Only 3 other items in the set use the `abs` function.	Create a new topic (combination of logic and arithmetic expressions).

analyses of item properties; these analyses provide insight into the potential applicability of studied properties. To illustrate the studied problem and potential solutions, we also describe two specific case studies: explainable outlier detection for English grammar items and interpretable clustering for Python programming items.

2. Related Work

Learning environments are a special type of software product; the principle of iterative improvement is a standard practice in software engineering, particularly in agile methodologies (e.g., Salo and Abrahamsson (2007)). Our work is not concerned with the improvement of the software code or the development process but rather with the improvement of the data used by the software. For this improvement, we aim to leverage the complementary strengths of human intelligence and machine learning techniques based on data analysis. Such techniques are denoted using the terms like *design-loop adaptivity* (Aleven et al., 2016), *intelligence augmentation* (Baker, 2016; Hassani et al., 2020), *integrating human and machine intelligence* (Doroudi, 2019), or *human-in-the-loop* (Zanzotto, 2019).

Another phrase closely related to this work is *closing the loop*; the general idea behind this phrase is to extend the standard research pipeline (learners → data → analysis) by a specific action that impacts the learners, thus creating a loop instead of a one-directional pipeline. The exact meaning of the phrase differs among authors. Clow (2012) focuses on methods that present the results of learning analytics directly to users of learning environments (learners, teachers, administrators). Another line of work closes the loop via the actions of content authors (Baker et al., 2007; Koedinger and McLaughlin, 2016; Liu and Koedinger, 2017). Studies of this type analyze data from a learning environment, identify specific topics that require revision, and then evaluate a modified version of the system to show whether the revision improved student learning. Multiple methods have been proposed to approach such redesign (Huang et al., 2021).

This type of study provides a principled approach to iterative improvement. However, these approaches have been applied only in rather limited settings and their scalability is challenging, as they require significant time and expertise. Practically used large-scale learning environments contain tens of thousands of practice items, divided into hundreds of topics; the management of this data is complex (Pelánek, 2020a). In our recent work (Pelánek and Effenberger, 2022), we made a general argument for the use of the "avoiding stupidity perspective" – rather than seeking optimal solutions, it may be more useful to focus on detecting clear defects. In this work, we elaborate on specific techniques that realize this perspective.

Several recent works have already proposed pragmatic approaches for the iterative improvement of large content using visualization and dashboards targeted toward system designers and content authors. Arruarte et al. (2021) propose a visual learning analytics tool targeted at authors of test-based exercises; the aim of the tool is to help authors improve the created materials. Effenberger and Pelánek (2021b) discuss methods for visualization of student-item interaction matrix; one of the applications of these visualizations is to facilitate the detection of deficient content. Pelánek (2021) provides an overview of visualization techniques relevant from a system designer's perspective.

Other works focus on automated heuristics that aim to locate defects. Mian et al. (2019) frame the problem of iterative improvement by the question "What's most broken?" and propose specific heuristics to answer the question. Fancsali et al. (2021) highlight the need to prioritize the redesign effort on the parts of the learning environment where the effort will have a high impact; they propose the problems of *targeting* and *focusing* design efforts and discuss several techniques for these problems. Fancsali et al. (2021) discuss techniques for targeting in more detail.

The specific aspect of this work is the focus on individual items. Most of the previous work focuses on the analysis and redesign efforts on a coarser level (topics, knowledge components, skills). One exception is the work by Darvishi et al. (2022) who analyze items in the context of learnersourcing. They describe a spot-checking algorithm that identifies items that would benefit from being reviewed by an instructor.

We also specifically focus on the interpretability and explainability of proposed techniques; this topic is recently getting increasing attention both in artificial intelligence in general (Linardatos et al., 2020) and in educational applications (Khosravi et al., 2022).

3. Problem Description

In this section, we provide an explicit formulation of the studied problem. To do so, we need to start by clarifying used terminology. We also describe a specific learning environment that is used as a source of data for our analysis and case studies.

3.1. Terminology

Research in the field of educational technology uses varied terminology (Pelánek, 2022). To make our discussion clearer, we start by clarifying some of the terms that we use.

We focus only on content that poses an answerable challenge to students; we do not consider educational materials that are used by students passively. We use a generic term *item* to denote any such content that can be answered and evaluated (e.g., questions, quizzes, multi-step problems). We focus mainly on multiple-choice questions; when discussing these, we use the terms *stem* (the main part of the question), *options* (the choices available), *correct answer*, and *distractor*. Figure 2 shows an example of such an item.

Items used within a learning environment are typically divided into some groups, which may be called, for example, knowledge components, skills, concepts, topics, or rules. In this work, we use a generic term *topic*. For example, the item in Figure 2 belongs to the topic "For loop in Python". We use the term *skill* to denote an estimate of a student's ability within some topic.

We distinguish the notions of *complexity* and *difficulty* (Pelánek et al., 2022). Difficulty measures are based on the performance data of students (e.g., the error rate), whereas complexity measures are metrics based on the definition of items (e.g., the length of the item statement).

Figure 2: Example of a multiple-choice question.

What does this code output?

```
for i in range(2):
   print("X")
   print("Y")
```

A)	*B)*
X	X
Y	X
X	Y
Y	Y

3.2. Problem Definition

Our aim is to identify items worthy of the attention of content authors. The attention-worthiness depends on a specific context and is at least partially subjective, so it is not feasible to fully formally define the problem. Nevertheless, it is useful to explicitly describe the expected inputs and outputs.

The inputs are:

- a large pool of items,
- domain model (item metadata), particularly mapping of items into sets of related items,
- data on student performance, particularly their answers and response times.

For the problem to be meaningful, the pool of items should be large enough that a periodic manual inspection of all items is impractical. Some techniques may be used even without data on student performance, but the absence of this data source severely restricts the potential of obtaining useful results.

The expected output is a subset of attention-worthy items together with information explaining the reason for attention-worthiness. The size of the output should be reasonably small so that it can be explored in a few hours at most. Sometimes it may be useful to present the results as clusters of items with the same explanation as this may facilitate the processing of results by content authors.

We consider an item *attention-worthy* if it is meaningful for the content author to spend time deliberating over the item. This deliberation should lead to a practical conclusion, typically one of the following:

- modifying items:
    - removing items,
    - updating items (e.g., changing their formulation, adding explanations),
    - adding new items inspired by the identified items,
- changing the domain model (e.g., splitting topics, adding new topics),
- obtaining insight – the content author gets some specific, useful insight applicable in the future preparation of new items.

The problem can be generalized to consider attention-worthiness not just for content authors but also for other human roles, particularly teachers. In this work, we consider only content authors in the context of design-loop adaptivity.

3.3. Data

The techniques that we discuss are relevant to any learning environment that uses a large pool of items (well-known examples of such systems are, for example, Khan Academy, DuoLingo, IXL, MATHia, or ALEKS). We use data and examples from the Umíme system (https://www.umimeto.org/). This system has a wide coverage of school subjects, including Czech (as a native language), English (as a second language), mathematics, computer science, biology, and geography. Practice within the system is publicly available without any registration with a daily limit. For unlimited access, either a school or individual license is necessary. The system is used by approximately 15 % of Czech schools; the number of daily users is in the order of tens of thousands. Most users are children between 8 and 15. Pelánek (2021) provides more details and examples of analysis that offer additional insight into the used data.

Within the used system, the content is divided into topics and assigned discrete difficulty levels (three values). The choice of specific items is given by a random sampling that takes into account the item difficulty levels and the difficulty setting chosen either by a student or a teacher. The length of practice is determined by a mastery criterion described in Pelánek and Řihák (2018). The properties of collected data thus reflect a mix of randomness, adaptivity, and selection biases. The exact properties of data are specific to a given system, but their basic characteristics are quite typical for this kind of learning environment.

Although the used system contains tens of thousands of items of several different types, the reported experiments primarily use items in the form of short multiple-choice questions. The main reason behind this focus is presentational – the use of multiple-choice questions, which tend to be short, allows us to use multiple illustrative examples throughout the paper. The presented techniques are applicable to more complex items as well (e.g., one of the other applications that we have done is the analysis of word problems in mathematics with constructed responses). The specific data about the used items are described in more detail in each experiment.

4. Item Properties

This section considers the "data processing" arrows in Figure 1. We provide an overview of item properties that may be useful for identifying attention-worthy items. We focus particularly on properties that share the following three characteristics: 1. They can be computed algorithmically. 2. They are applicable to a wide range of items from various domains. 3. They are easily interpretable by content authors.

4.1. Properties based on Content Data

At first, we consider item properties that are based only on the content data, i.e., these properties can be computed even before the item is applied in the learning environment. To illustrate the measures, we will use a multiple-choice question presented in Figure 2.

4.1.1. Complexity Measures

Complexity measures provide a quantitative summary of an item. The most elementary one is the length of the text in the item. More nuanced data are provided by text complexity measures, either by readability formulae (Benjamin, 2012) or by more complex text analysis (Sheehan et al., 2014).

Beyond the general text complexity measures, it is possible to devise complexity measures specific to a particular domain (Pelánek et al., 2022). In programming, items (or their solutions) often contain code fragments and we can use code complexity measures (e.g., number of lines, nesting depth, cyclomatic complexity).

For the example in Figure 2, we could measure the length of the text (27 characters), code (46 characters including whitespace), or the total length including both options (81 characters).

We could summarize the complexity of the code as the number of lines (3 lines), the depth of nesting (2 levels), or the cyclomatic complexity (1 decision point, which corresponds to the cyclomatic complexity of 2).

4.1.2. Item Form

Other item properties are qualitative. For example, items differ in the included modalities (text, image, audio); we call this property *item form*. For multiple-choice questions, the form specifies the modality of the stem and the options. Some modalities are only applicable to the options (e.g., number, true/false). Additionally, some modalities are domain-specific (e.g., mathematical expression, chemical reaction, programming code).

To illustrate some decisions involved in the form specification, consider the item in Figure 2. The stem contains two modalities (text and code). In such a case, we might decide to keep just the more specific one (code). The options are texts, but we might also decide to be more specific and distinguish *code output* as a separate modality.

4.1.3. Item Type

Another property of items is the cognitive process they entail, such as recalling, computing, or constructing. For instance, the item in Figure 2 asks the students to *compute* the output of a given program, while another item could ask them to *construct* a program that produces a given output.

A useful list of common cognitive processes is given by the revised Bloom's taxonomy of educational objectives, specifically its cognitive process dimension (see (Anderson and Krathwohl, 2001) for a detailed description of the taxonomy or (Krathwohl, 2002) for a brief overview). The taxonomy is hierarchical, with 6 broad categories (remember, understand, apply, analyze, evaluate, create) and about 20 more specific subcategories. For example, the category *understand* includes processes such as exemplify, classify, summarize, compare, and infer.

While the actual cognitive processes are non-observable and may differ among students, it is possible to assign each item the dominant cognitive process based just on its content. Previous research has demonstrated that items can be automatically classified to one of the broad categories in Bloom's taxonomy using either rules over verbs and other keywords (Omar et al., 2012) or more complex machine learning models over words (or their embeddings), possibly weighted by a TF-IDF variant (Mohammed and Omar, 2020; Yahya et al., 2012). However, as a given item set might have only two or three of the broad categories, we might need a classification into the subcategories. In section 6.2, we present a semi-automatic method to classify a large number of items using regular expressions.

4.1.4. Content Representation

We can also compactly encode the content of the item. The basic approach is the use of the bag-of-words representation or vectors of TF-IDF weights for the most informative terms. It is also possible to apply specialized techniques for the automatic identification of keywords in items; Goutte et al. (2015) provide a specific proposal.

For example, in programming questions, we can automatically determine the used keywords and commands (e.g., `for`, `range` and `print` in the example in Figure 2). Below, we report an analysis of English grammar items where we utilize properties based on rankings of words used in the item text, specifically their Common European Framework of Reference (CEFR) levels.

4.2. Properties based on Performance Data

Now we consider item properties that are based on the observed data about student performance. The advantage of these properties is that they are widely applicable—their computation is not specific to any particular domain.

In this work, we utilize only simple descriptive properties of student performance. These simple measures have disadvantages: they do not take into account the specific population of students who answered the item (i.e., they are prone to be influenced by a selection bias) and they do not take into account the order in which students answered items (which may bias the data when students are learning during the practice). These disadvantages can be overcome by the use of student modeling approaches, e.g., item response theory models (Baker, 2001), Elo rating system (Pelánek, 2016), or various variants of knowledge tracing models (Pelánek, 2017). However, even these approaches retain some simplifying assumptions, e.g., item response theory does not take learning into account. Moreover, although these approaches may provide a more complex analysis of one aspect of student performance (the correctness of answers), they often completely ignore other aspects (response times or specific values of wrong answers). For the purpose of identifying attention-worthy items, it may be more useful to use a wider range of simple measures since items may be attention-worthy for many different reasons[1].

4.2.1. Difficulty Measures

A key aspect in finding items worthy of attention is clearly difficulty—items that are too difficult or too easy are natural candidates for revision. The basic difficulty measures are *error rate* (the proportion of students that made a mistake) and *median response time*[2].

4.2.2. Item Discrimination

A discrimination index tells us how good the item is in differentiating between students with a high level of measured skill and those with a low level of skill. For this work, we have explored two different approaches for calculating item discrimination associated with the classical test theory (Brennan, 1972; Oosterhof, 1976; Pyrczak, 1973).

The simplest measure of item discrimination is the *upper-lower discrimination index* (Brennan, 1972). Consider a set of students who answered item i. We sort these students with respect to their overall performance on other items and take the upper third U and the lower third L. The discrimination index is the difference between the error rate on the item i of students in set L and set U.

The second simple method of discriminant calculation is the *point-biserial index*, which is given as a correlation coefficient between the overall error rate of students (continuous variable) and their answer on the item i (dichotomous variable) (Kavitha et al., 2012).

4.2.3. Performance-based Item Similarity

An item can be unsuitable not because of its inherent features but because it does not fit the context of other items. To check this aspect, we can measure the similarity of items based on student performance (Pelánek, 2020b). Specifically, we consider the average similarity $avgsim$,

[1] The items in education are subject to the Anna Karenina principle that states that while no feature guarantees success, many guarantee failure (Diamond, 1998; Kirilenko et al., 2021).

[2] Response times typically contain outliers, so median is a better measure of central tendency than mean.

which is computed as follows: for each pair of items i, j, we compute sim_{ij} as the Pearson correlation over responses (by students who answered both items); $avgsim_i$ is the average over sim_{ij}. For this analysis, we consider only binary responses (correct, incorrect).

4.2.4. Common Wrong Answers

When students can choose from several options or have to construct a response (e.g., mathematics examples with written answers), it may be useful to analyze not just the error rate but also the specific values of wrong answers. Previous work has shown that the distribution of wrong answers is typically uneven, i.e., there often exists some common wrong answer (e.g., Pelánek et al. (2016)). The specific value of the common wrong answer is valuable feedback to content authors and may serve as a specific impulse for revision for identified items. In the analysis below, we explore whether the numerical rate of the common wrong answer may be useful for identifying attention-worthy items.

5. Analysis of Item Properties

Before trying to apply the item properties for the identification of attention-worthy items, we perform their analysis to explore their usefulness and usability.

5.1. Reliability of Measures based on Performance

To use properties based on student performance data, we need to know whether they are stable. We report two experiments that try to answer the following question: How much data do we need for the properties based on student performance to be reliable?

5.1.1. Split-half Reliability Experiment

The first experiment is a split-half reliability experiment. Split-half reliability is typically used in psychometrics to evaluate the reliability of some person trail measures; see Steinke et al. (2021) for a typical application or Beck (2005) for an application in educational data mining. We use the experiment to evaluate the reliability of item parameters; see Rachatasumrit and Koedinger (2021) for a similar application of the experiment.

Specifically, we split the available student data into two independent halves, compute the item properties based on each half and analyze the agreement of resulting values. For a measure to be reliable, we should obtain similar values from both halves of the data.

We report results for multiple-choice questions in the domain of English grammar. We perform the analysis separately for several topics as this gives us insight into the generalizability of the results. We selected the most popular topics within the system. These topics have over 1000 answers for each item. The number of items in each topic is between 130 and 180.

We use several subsamples of the available data to explore how the reliability increases with the amount of data. We split the data into halves by the parity of student identification number. To measure the agreement, we use the Spearman correlation coefficient of values computed on the two halves. We have chosen the Spearman correlation coefficient because, for our purposes, we are interested primarily in the ordering of values. We have performed the analysis also with respect to other measures of agreement (e.g., mean absolute deviation or the agreement on the top 10 items); they lead to similar conclusions.

Figure 3 reports the results. We use “the number of answers per item” as a primary measure of the size of needed data. The results show that for the basic difficulty measures (the error rate, median response time), we mostly obtain good reliability ($r \sim 0.8$) with 200 answers per item and nearly perfectly stable ordering ($r \sim 0.95$) with 500 answers per item. As the figure shows, there are slight differences between various topics.

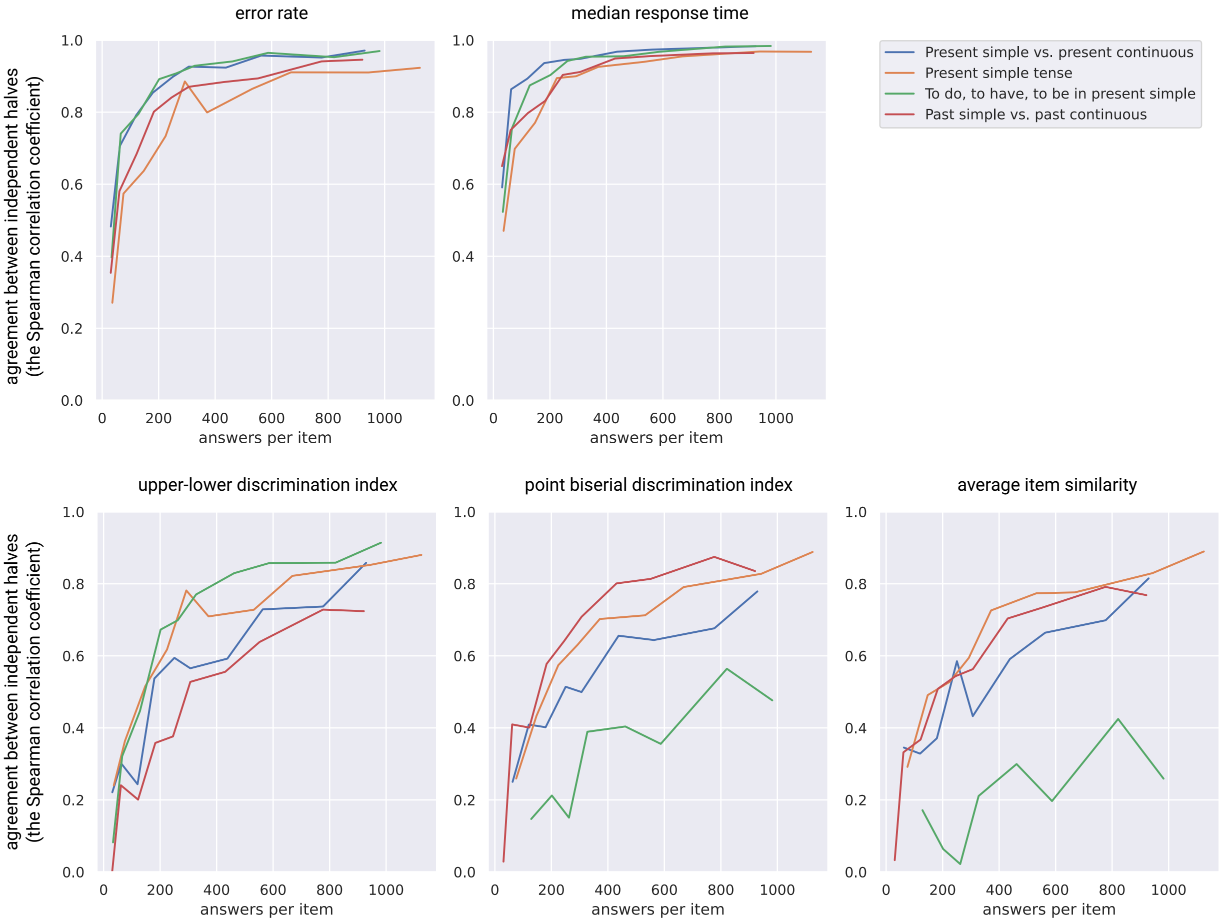


Figure 3: Reliability of five item properties that are based on student performance.

Discrimination indices and average similarity require more data. Their convergence is much slower. With 500 answers per item, the values have reasonable reliability, but they are far from perfectly stable even with 1000 answers per item. Moreover, the results show quite large differences among topics. This suggests that any use of these indices would require extensive data on student answers and careful analysis of their reliability for the particular data set.

5.1.2. Simulation

The previous analysis is based on data from a specific system. This has the advantage of direct relevance to applicability in a real use case. However, it may be that the results are influenced by some specific aspect of the collected data, e.g., attrition bias in data collection (Pelánek, 2018). We complement this analysis with experiments based on simulated data. In this way, we explore an idealized scenario, which gives us best-case estimates for the amount of necessary data.

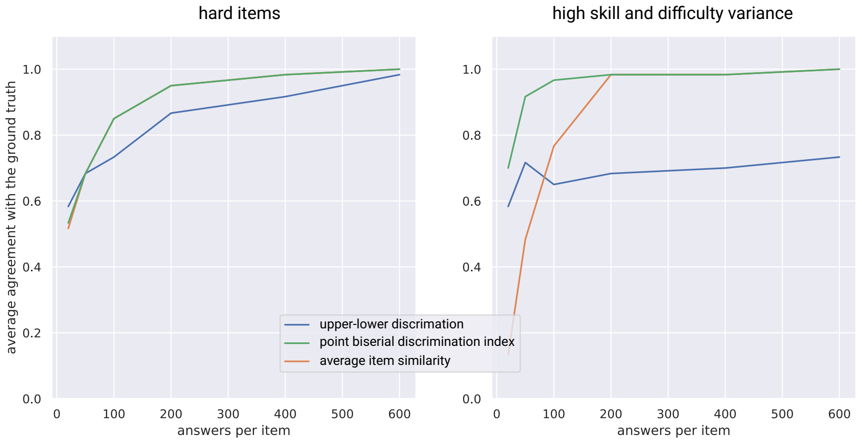


Figure 4: Agreement with the ground truth in experiments for simulated data. Note that in the left graph, the orange line is mostly overlapped by the green one.

To generate simulated data, we use the Additive factors model (Cen et al., 2006; Durand et al., 2017) – a simple logistic model of learning that supports multiple knowledge components. We use a slight extension of the model. Standardly used Additive factors model has difficulty parameter only per knowledge component, while we use difficulty parameter per item. This better corresponds to observed data and the studied research question. We use a model with two knowledge components; student skills within these components are independent. The simulated system contains 50 items; 47 of these items belong to the first knowledge component, 3 of them belong to the second knowledge component. These 3 items are considered to be the ground truth outliers—the items that we want to identify as worthy of attention. We generate the simulated answers based on the probabilities provided by the additive factors model; all simulated students answer all items in random order.

Given the generated data, we compute the discrimination indices and average similarity of items. For each index, we take the 3 items with the lowest value and evaluate the agreement with the ground truth (the 3 items from the second knowledge component). We report the average agreement over 20 runs. Figure 4 shows the results for different numbers of simulated students. The exact results depend on the specific setting of the simulation parameters (e.g., speed of learning, mean item difficulty, the variance of student skills and item difficulties). We show results for two settings of parameters. The source code of the experiment is available[3] and allows exploration of other settings.

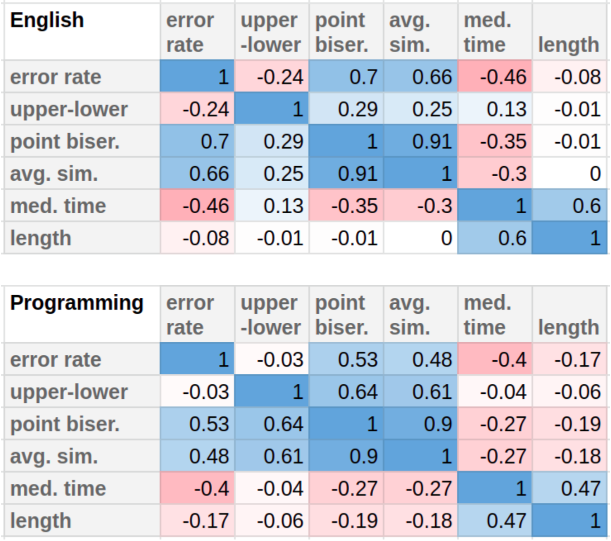
This is an optimistic scenario for the application of the studied indices as there are no missing data, no biases, and the misfitting items are perfectly orthogonal to others. The results show that even in this optimistic scenario, we need at least 200 answers per item to obtain stable performance. Particularly the upper-lower index is weak; in some settings, it is not able to reliably identify the misfitting items even with large data (particularly in the presence of very easy items).

[3] https://github.com/adaptive-learning/jedm-item-revisions

5.2. Agreement between Properties

As the next step, we analyze relations among item properties. When two of the item properties are strongly correlated, one of them can be omitted as it does not bring useful additional information. Such pruning simplifies the subsequent analysis and interpretation of results. With weaker correlations, we may want to keep both properties. In both cases, the analysis can give us insight that is useful for the interpretation of computed values.

Figure 5 provides summary results for the agreement of several widely applicable item properties. The results are reported for multiple-choice questions in English and programming. In each case, we have used 10 topics with the most answers (over 750 thousand answers for each topic in the case of English, over 25 thousand answers for each topic in the case of programming). There is, as can be expected, a positive correlation between item length and median response time. The error rate is negatively correlated with response times.



English	error rate	upper -lower	point biser.	avg. sim.	med. time	length
error rate	1	-0.24	0.7	0.66	-0.46	-0.08
upper-lower	-0.24	1	0.29	0.25	0.13	-0.01
point biser.	0.7	0.29	1	0.91	-0.35	-0.01
avg. sim.	0.66	0.25	0.91	1	-0.3	0
med. time	-0.46	0.13	-0.35	-0.3	1	0.6
length	-0.08	-0.01	-0.01	0	0.6	1

Programming	error rate	upper -lower	point biser.	avg. sim.	med. time	length
error rate	1	-0.03	0.53	0.48	-0.4	-0.17
upper-lower	-0.03	1	0.64	0.61	-0.04	-0.06
point biser.	0.53	0.64	1	0.9	-0.27	-0.19
avg. sim.	0.48	0.61	0.9	1	-0.27	-0.18
med. time	-0.4	-0.04	-0.27	-0.27	1	0.47
length	-0.17	-0.06	-0.19	-0.18	0.47	1

Figure 5: Agreement between properties (correlation averaged over the top 10 most solved topics).

5.2.1. Discrimination Indices

The results in Figure 5 are averaged across topics. These summary averages sometimes mask non-trivial differences that are present in the results for individual topics. This is particularly the case for the upper-lower discrimination index. This index is, in some cases (e.g., *To do, to have, to be in present simple*), strongly correlated with the error rate, whereas in other cases (e.g., *Past simple vs. past continuous*), we see a negative correlation. There is also a high variation among topics in the relations between the upper-lower discrimination index and the point-biserial discrimination index. They both quantify the same aspect ("how well does the item discriminate") but in quite different manners. The difference is marked particularly for easy items, where the upper-lower index always produces low discrimination values.

Interestingly, we find a very strong correlation between the point-biserial discrimination index and average similarity. At first sight, this seems surprising, as these two item properties capture intuitively different aspects of items: how well does the item discriminate between good and poor solvers versus how is the item similar to other items. However, discrimination and similarity are linked—poor discrimination of an item means that the item behaves differently than other items in the same topic.

Although the average similarity and the point-biserial discrimination index have different intuitions behind them, they are actually mathematically defined in a similar fashion. In the computation of average similarity, we compute Pearson correlation (over dichotomous variables) and then perform averaging (for each item, over items). In the computation of discrimination as point-biserial correlations, we do averaging (for each student, over items) to compute skill estimate and then perform Pearson correlation (over one continuous and one dichotomous variable). Both computations thus involve the Pearson correlation coefficient and averaging; they differ in the order of these operations. Although these two operations do not commute, from a practical perspective, the results are nearly the same (at least on our data).

5.2.2. Error Rate and Common Wrong Answers

For some types of items, students construct an answer and thus may produce different wrong answers. Is it useful to analyze these mistakes and their prevalence? The answer depends on the particular situation. The distribution of wrong answers depends on the specific topic. In some cases, the prevalence of the most common wrong answer is over 50% of all wrong answers. This is the case, for example, of *Expressions with absolute value* or *Prepositions of time*. In other cases, the distribution is much flatter—this we see, for example, in the case of *Expressions with fractions*, where there are more potential sources of a mistake. Nevertheless, even in these topics, the most common answer comprises, on average, 10% of all wrong answers.

For a few topics, the rate of the most common wrong answer is closely correlated with the error rate. In these cases, it may be useful to use the specific value of the answer as an impulse for content authors, but the value of the rate does not bring much information useful for the automatic detection of attention-worthy items. For many topics, however, the correlation is weak, and thus the rate of the common wrong answer provides additional information. Specific examples are provided in Figure 6. For example, although the examples $2z - (2z + 1) - 1$ and $(10 - 7)(2a - 4)$ have a very similar error rate, the first one has a very common wrong answer (0), whereas for the second one, there are many different wrong answers.

6. Case Studies

In this section, we report two case studies where we analyze item properties and identify items for revision. In the first case study, we utilize a standard outlier detection technique which we extend with post-processing to obtain explainable results. In the second case study, we use an interpretable clustering technique tailored specifically toward this particular application.

6.1. Explainable Outlier Detection

In the first case study, we use a standard outlier detection technique, which we extend with explanations that should facilitate the interpretation of results by content authors. For this case study, we used English grammar items: 6280 items, covering 65 topics, typically 97 items per topic. All items are multiple-choice questions with two options (correct answer and distractor). The distribution of student performance data, which are used to compute difficulty measures, is

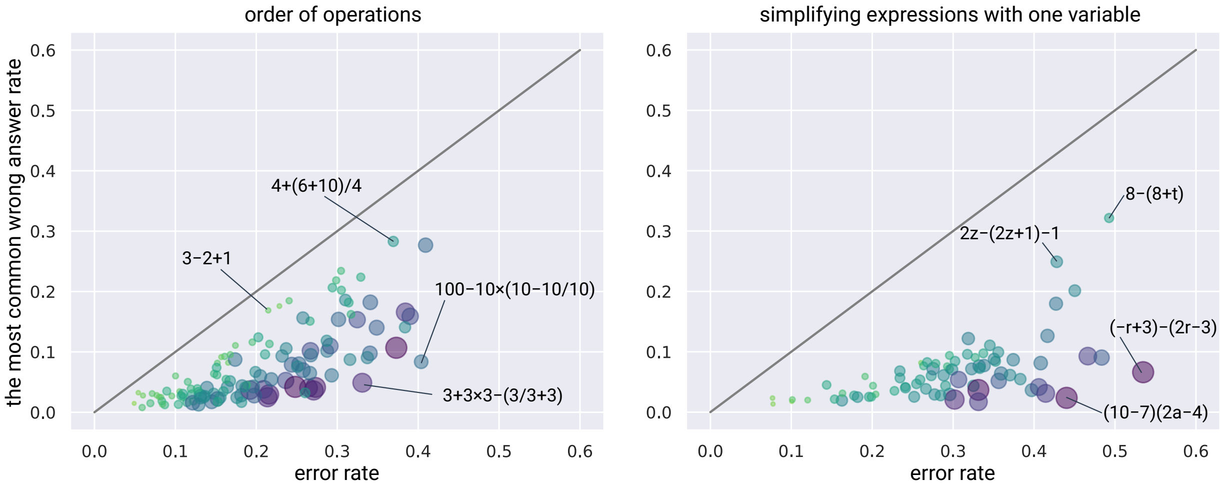


Figure 6: The error rate and the rate of the most common wrong answer. Size and color are proportional to response time (larger and darker = higher).

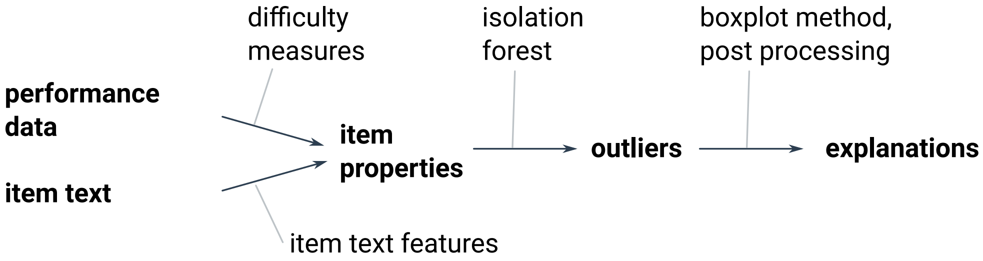


Figure 7: Explainable outliers for English grammar items – processing pipeline.

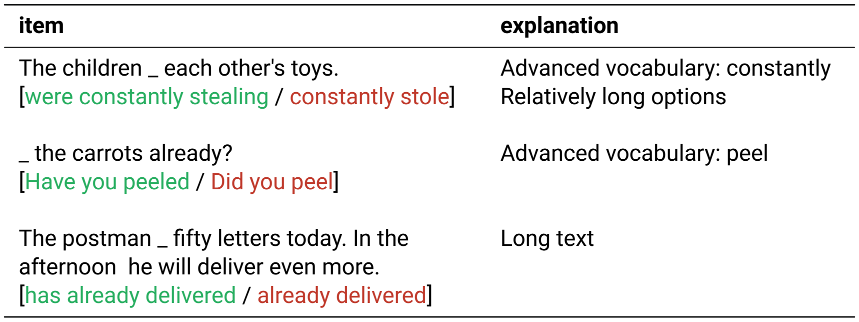
highly uneven. Some items were answered by tens of thousands of students, others only by a hundred.

Figure 7 illustrates the processing pipeline used. We used both the item text and student performance data to compute item properties. Over the computed properties, we run an out-of-box outlier detection algorithm. For the detected outliers, we then construct simple explanations.

As item properties based on performance data, we have used the two basic difficulty measures (error rate and median response time). As item properties based on item content, we have used several types of properties:

- Syntactical features: the item length, the ratio of the stem length and the answer length, the ratio of the answer to distractor length, and Levenshtein distance between the correct answer and distractor (normalized by their length).

- Vocabulary features. To compute these features, we use the following process: We take the item text (e.g., *You seem distracted. What _ about? [are you thinking / do you think]*), take its bag-of-words representation, which we lemmatize, and to each word we assign its (numerically encoded) CEFR level (*you*→*1, seem*→*2, distract*→*4, what*→*1, be*→ *1,* . . .), and based on the vectors of levels we compute quantitative properties (*max. level = 4, average level = 1.36, the ratio of words on level 1 = 0.82,* . . .). To assign CEFR



item	explanation
The children _ each other's toys. [were constantly stealing / constantly stole]	Advanced vocabulary: constantly Relatively long options
_ the carrots already? [Have you peeled / Did you peel]	Advanced vocabulary: peel
The postman _ fifty letters today. In the afternoon he will deliver even more. [has already delivered / already delivered]	Long text

Figure 8: Explainable outliers: illustrations of results.

levels to words, we used a combination of three existing resources[4]. The CEFR levels are commonly denoted A1, A2, B1, B2, C1, and C2. We used numerical encoding (1 to 6).

- Part-of-speech features: binary indicators of the presence of a noun, a verb, and an adjective in the item text.

We run the outlier detection separately for each topic (e.g., 247 items belonging to *Present simple vs. present continuous*). Within such a set of items, we search for global outliers. We use Isolation forest (Liu et al., 2012), which is a technique suitable for such situations with good performance results (Emmott et al., 2013). We use the scikit-learn implementation (Pedregosa et al., 2011).

Outlier detection algorithms compute an outlier score for each data point. To report results, we need to set a threshold on the score. One approach is to set a contamination parameter (a ratio of points reported as outliers). This is not suitable in our setting since, in some topics, there may be no interesting outliers, whereas in others, there are several. We have used a fixed threshold value to report outliers (-0.6 for the isolation forest implementation). This value provided a reasonable number of outliers.

For explanations, we experimented with several variants, including explanations in the form of 2D scatter plots as proposed in the LookOut algorithm (Gupta et al., 2019). We decided to use a simple listing of properties with extreme values. To decide whether an item property has extreme value, we use the basic boxplot method (interquartile range, 1.5 multiple). Finally, we use simple post-processing that groups several related properties and assigns them easily understandable labels ("advanced vocabulary," "long item/time," "difficult item," "relatively long option"). Figure 8 shows an illustration of the results. The final result has 120 lines of this type.

Experience from this case study shows that even the relatively direct application of a general outlier detection technique (without intensive tuning of features or parameters) leads to the identification of items that deserve attention. We presented the result to the content author, who was not involved in the method development in any way. The results and provided explanations are useful, even when concise and simplified. A minor problem is that in some cases the results are a bit confusing due to the fact that the outlier detection method works over multidimensional data, whereas the explanation mentions only a single dimension ("Why was this item identified, whereas the other one which seems more extreme was not?").

[4] Oxford 5000 (https://www.oxfordlearnersdictionaries.com/wordlists/oxford3000-5000), English profile (https://www.englishprofile.org/wordlists/evp), CEFR-J (https://www.cefr-j.org/)

Among the 120 identified items, most are identified due to the presence of advanced vocabulary and the corresponding action to take is typically “remove the item” or “replace the difficult word with another one.” One reason why identified outliers are mostly with respect to advanced vocabulary may be that difficulty measures (error rate and median response time) have already been used in an ad-hoc manner in previous iterations of content development. Consequently, the main outliers with respect to them have already been removed.

The dominance of “advanced vocabulary” detections may, however, also be due to the specific choices that we made (e.g., the use of multiple features related to vocabulary). On the one hand, the results of outlier detection are quite sensitive with respect to the choice of features, specific algorithms, and parameter values. On the other hand, some items were detected quite robustly (e.g., *My cat [is/are] grumpy.* is quite a robust outlier due to the presence of the word “grumpy,” which is a significant outlier within a basic set on present simple tense). It may be useful to develop techniques that quantify this sensitivity and robustness of results.

6.2. Interpretable Clustering

Another use of the item properties is to cluster similar items. Clusters that contain few items or deviate considerably from the other clusters in a property are worth attention. For interpretability, we need to accompany the clusters with a concise description. The description should apply to all items in the cluster (perfect recall), but ideally *only* to the items in this cluster (perfect precision) (Effenberger and Pelánek, 2021a).

There are many clustering algorithms, but most of them are not interpretable. The simplest approach to interpretable clustering is to group items that agree on a small number of manually selected informative properties, such as the type and form of the item. Using too many properties (e.g., the presence of individual words in the question) would lead to all clusters containing just a single item. To create non-trivial clusters, we would need a more sophisticated interpretable clustering algorithm. One approach is to describe clusters using branches in a decision tree (Basak and Krishnapuram, 2005; Dasgupta et al., 2020; Liu et al., 2000); another is to select combinations of properties that frequently occur together (Effenberger and Pelánek, 2021a; Saisubramanian et al., 2020).

In this case study, we group items that share the same type, form, and size. We will see that even this simple grouping based on exact matching can provide valuable insight if the chosen properties are informative and appropriately granular to get non-trivial clusters. We used 1025 binary choice items, covering 8 introductory Python programming topics (*Variables and numerical expressions*, *Logic expressions*, *If*, *For*, *While*, *Functions*, *Strings*, *Lists*). Each topic contains between 80 and 212 items. The reported difficulty measurements are computed from 597,367 collected attempts, which amounts to 583 students per item on average (minimum 30, maximum 4219). Figure 9 illustrates a sample output of the clustering for the topic *Logical expressions*.

6.2.1. Item Properties for Clustering

The items are grouped by three properties: the type (based on the cognitive process), the form (based on the modalities of the stem and options), and the size (based on the number of characters together in the stem and the options). We discuss these content-based properties in Section 4; here, we only mention additional decisions needed for this specific use case.

We use 7 types of items: recognize, compute, construct, summarize, compare, transform, and infer. We considered other cognitive processes described in the second level of Bloom's taxonomy (Krathwohl, 2002), but only these 7 were present in this particular topic. To automatically detect the types, we have iteratively developed a set of regular expressions over the item formulation until we covered all items. Each type required multiple patterns; for instance, the *compute* would match on *What does this (code|program) outputs\?* but also *How many .* will be printed\?*

Concerning the form of the item, we included two general modalities (text, yes/no options) and two domain-specific modalities (code, output). Code includes single-line expressions and code fragments. Output includes values (numbers, strings) and printed text. We abbreviate the form as *x→y*, where x is the first letter of the most specific modality in the stem and y is the first letter of the options' modality. For example, *c→o* means that the stem includes code and the options are outputs.

Concerning the size, we discretized it into three bins in order to have a reasonable total number of clusters. The thresholds—60 characters for small items and 100 characters for large items—were selected such that approximately a quarter of all items in the sets practicing introductory programming is small and a quarter is large.

A group of items is then specified by the item type, form, and size, using abbreviations listed in Figure 9. For example, *con/t→c/L* are large items that ask the students to construct a code from a text description.

6.2.2. Clustering Visualization

In Figure 9, the clusters are located according to their mean error rate and response times.[5] Such a plot allows to quickly identify attention-worthy clusters that deviate in difficulty. These outliers, unless in a large cluster of many items, would be already spotted in a simple difficulty scatterplot of items, but the cluster descriptions explain the difficulty by the interpretable content-based properties. For example, there is a cluster of three *compare* items with extremely high response times at the top of Figure 9, but there are other shorter *compare* questions with more moderate response times, which suggests that *compare* questions might be adequate if they are kept short.

Even the clusters that do not deviate in difficulty are worth attention if they contain just a few items. To increase homogeneity, such outlying clusters could be removed, but since they have appropriate difficulty, a better solution might be to create more items with the same properties (type, form, and size in our case). For example, there are only one small and two medium items to construct code from text description (*con/t→c*). We might decide to add other items to practice this constructing skill, but we should avoid long questions and, even then, not expect the items to be particularly easy.

6.2.3. General Trends

Exploring and comparing such visualizations for several topics can help the content authors to better understand the composition of the topics and the causes of difficulties. The difficulty depends not just on the tested topic but also on the cognitive processes, which are represented by the question types. In our context, the questions to summarize, compare, transform, and infer

[5] We use the median to aggregate response times *within* an item and then the mean to aggregate these medians *across* items.

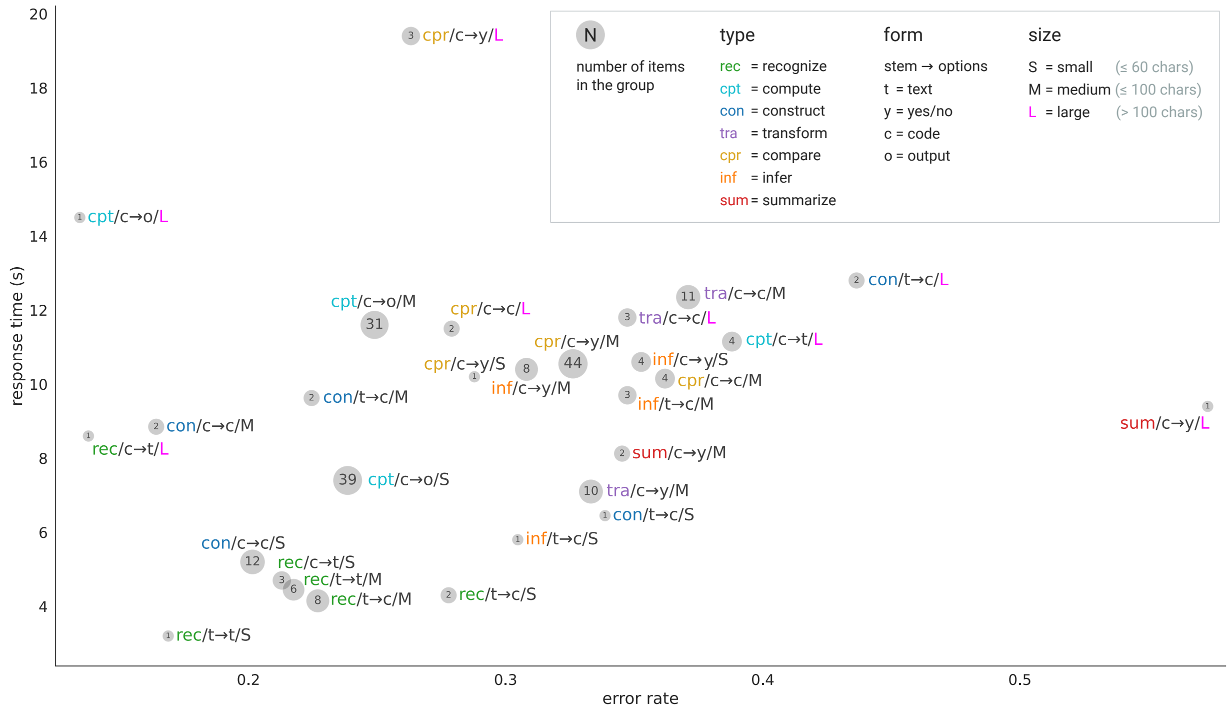


Figure 9: Item clusters in *Logical expressions in Python*.

are generally more difficult than the questions to construct and compute, which are, in turn, more difficult than the questions to recognize a basic fact about the language. Note that the most difficult questions in our data (such as *infer* and *summarize*) would be classified as the second level of Bloom's hierarchy (*understand*), while *construct* and *compute*, although easier, would fall into the third level (*apply*). Consequently, using just the coarse levels of Bloom's hierarchy would not help to illuminate the causes of the difficulties.

The difficulty variation due to the involved topic and cognitive processes is typically intrinsic to the practiced topic, so non-homogeneity would be preferably resolved by splitting the topic into multiple ones rather than by removing items. In contrast, some difficulty variation can be linked to extraneous factors and these we would like to eliminate. We have observed that the undesirable difficulty in the programming questions could be typically attributed to one of three sources, each linked to one of the three inspected properties.

- The first and the most obvious one is the excessive length, which is also apparent from Figure 9.

- The second source is an additional level of indirection in the item options. The questions to compute a result become more difficult if the options are not concrete outputs (*cpt/c*$\rightarrow$*o*) but rather just a text description (*cpt/c*$\rightarrow$*t*).

- The third source is a mismatch between the complexity of the presumed cognitive process and the complexity of the targeted knowledge. Consider the question "Can `not x != y` be simplified to `x = y`?" Seemingly, the item practices the skill of applying negation to simplify a condition, but it covertly tests the knowledge of basic syntax (= vs. ==).

Students are primed by the question formulation to focus on the logic and do not attend to syntax details. This is, certainly, a tricky question, as evidenced by the error rate of 61%.

7. Discussion

To conclude, we recapitulate the main task that we proposed in this work, summarise recommendations for practice, and discuss the limitations of the current work and directions for future research.

7.1. Attention-Worthy Items and their Interpretation

One of the main goals of this paper is to highlight the task of identifying items worthy of the attention of content authors. We have to acknowledge that it is a fundamentally ill-defined problem. The attention-worthiness depends on a specific context and is at least partially subjective. It is not possible to provide a universal definition of a "good" or "bad" item. Nevertheless, this task is practically very useful and it can be addressed with the use of educational data mining techniques. It thus deserves attention.

Although we stressed the focus on individual items, the problem is not clearly demarcated. The quality of items is inherently interlinked with the quality of the domain model. An item may be attention-worthy because of poor domain modeling, e.g., the insufficient granularity of used knowledge components. In these cases, the suitable action is not to change the item itself but rather to improve the domain model. The use of techniques for the identification of attention-worthy items thus overlaps with the use of techniques for revision of domain models like Liu and Koedinger (2017) and Fancsali et al. (2021). For this work, we purposedly highlighted the focus on individual items. For future progress, it may be useful to consider the relation to domain modeling in more detail.

7.2. Recommendations for Practice

Although the presented results are mostly exploratory and do not provide definitive evaluation, they do provide basic guidance for practical applications. Our analysis (particularly the reliability analysis in Section 5.1) and practical experience with performing item revisions suggest that the primary tool for identifying attention-worthy items are the basic performance properties: the error rate and median response time. The properties are simple yet practically very useful. They are widely applicable, have intuitive interpretation, and are reliable even with relatively small data (at least compared to discrimination indices). Even though these simple properties can be influenced by biases in data, this is probably not a major issue when the aim is to identify problematic items. Although there is clearly space for the use of more complex item properties, the usage of error rate and median response time is a good starting point.

Discrimination indices and average item similarity are potentially useful techniques that can sometimes identify attention-worthy items that would not be detected using basic difficulty measures. To be reliable, these methods require larger data (at least 200 answers per item). Our experiments suggest that the upper-lower discrimination is not sufficiently reliable for practical usage. The point-biserial index and average item similarity are highly correlated and provide better insight (when enough data are available).

Specific attention-worthy items can be identified using out-of-the-box outlier detection techniques. However, the obtained results can be quite sensitive to the specific choice of an algorithm and its parameters. To get better insights, it is useful to construct explanations that describe the main reason why an item is an outlier or to develop more specific, interpretable analysis tools that are tailored specifically to the task of identifying attention-worthy items.

7.3. Limitations and Future Work

In this work, we use only simple item properties that do not take into account student learning or biases present in data (attrition bias, selection bias, ordering of items), which can be caused by student behavior or system behavior (e.g., adaptive selection of items) (Pelánek, 2018).

This can be overcome by the use of more advanced modeling methods, for example, using Item response theory models (Baker, 2001) or student modeling techniques like the Additive factors model or Bayesian knowledge tracing (Pelánek, 2017). However, even these methods still make many simplifying assumptions. Item response theory models take into account differences in student skills and include a discrimination parameter, but they assume no learning. Additive factors model and Bayesian knowledge tracing model incorporate learning (in some specific form) but do not contain difficulty or discrimination parameters (per individual items). Consequently, none of these standard models is optimal for the purpose of identification of attention-worthy items in learning environments.

Based on our exploration, we believe that for the immediate practical application, it is more fruitful to use a wide range of simple item properties rather than delve deeply into one of them. There are certainly cases where the biases in data could lead to misleading conclusions and where the usage of more principled approaches would be warranted. Further explorations into these issues constitute an interesting research topic for future work.

In our analysis, we have provided a more detailed analysis only for textual items. Items in learning environments often contain other types of data, particularly images and animations. The content analysis of these types of items is more challenging. Identification of attention-worthy items with multimedia data is thus another interesting research direction.

We have used a wide range of topics from several educational domains (mathematics, programming, English as a second language). Although the basic trends and technique applicability are the same across domains, the specific results often show non-trivial differences in results even for seemingly similar topics. For example, the correlation between the error rate and the upper-lower discrimination index ranges from strongly positive to strongly negative. In the current work, we point out several such cases. We do not, however, provide an explanation for these differences or recommendations for the practical treatment of these differences. This topic also deserves more attention.

In the present study, we have focused on explicit problem formulation and a broad overview of possible approaches that can be used to tackle it. The presented analysis is mostly exploratory. For future progress, it will be useful to perform thorough evaluations of specific techniques for identifying items worthy of revision.

References

Aleven, V., McLaughlin, E. A., Glenn, R. A., and Koedinger, K. R. 2016. *Handbook of research on learning and instruction*. Routledge, Chapter Instruction based on adaptive learning technologies, 522–559.

Anderson, L. W. and Krathwohl, D. R. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

Arruarte, J., Larrañaga, M., Arruarte, A., and Elorriaga, J. A. 2021. Measuring the quality of test-based exercises based on the performance of students. *International Journal of Artificial Intelligence in Education 31,* 3, 585–602.

Baker, F. B. 2001. *The basics of item response theory*. ERIC.

Baker, R. S. 2016. Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education 26,* 2, 600–614.

Baker, R. S., Corbett, A. T., and Koedinger, K. R. 2007. The difficulty factors approach to the design of lessons in intelligent tutor curricula. *International Journal of Artificial Intelligence in Education 17,* 4, 341–369.

Basak, J. and Krishnapuram, R. 2005. Interpretable hierarchical clustering by constructing an unsupervised decision tree. *IEEE transactions on knowledge and data engineering 17,* 1, 121–132.

Beck, J. E. 2005. Engagement tracing: using response times to model student disengagement. In *Proceedings of Artificial intelligence in education: Supporting learning through intelligent and socially informed technology*, C.-K. Looi, G. McCalla, B. Bredeweg, and J. Breuker, Eds. IOS Press, 88–95.

Benjamin, R. G. 2012. Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review 24,* 1, 63–88.

Brennan, R. L. 1972. A generalized upper-lower item discrimination index. *Educational and Psychological Measurement 32,* 2, 289–303.

Cen, H., Koedinger, K., and Junker, B. 2006. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of Intelligent Tutoring Systems*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds. Springer, 164–175.

Clow, D. 2012. The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, S. B. Shum, D. Gasevic, and R. Ferguson, Eds. Association for Computing Machinery, 134–138.

Darvishi, A., Khosravi, H., Sadiq, S., and Gašević, D. 2022. Incorporating AI and learning analytics to build trustworthy peer assessment systems. *British Journal of Educational Technology 53,* 4, 844–875.

Dasgupta, S., Frost, N., Moshkovitz, M., and Rashtchian, C. 2020. Explainable k-means clustering: Theory and practice. In *XXAI: Extending Explainable AI Beyond Deep Models and Classifiers, ICML Workshop.*

Diamond, J. M. 1998. *Guns, Germs, and Steel: the Fates of Human Societies*. W. W. Norton & Co.

Doroudi, S. 2019. Integrating human and machine intelligence for enhanced curriculum design. *PhD diss., Air Force Research Laboratory*.

Durand, G., Goutte, C., Belacel, N., Bouslimani, Y., and Leger, S. 2017. Review, computation and application of the additive factor model (AFM). Tech. rep., Tech. Report 23002483. National Research Council Canada.

Effenberger, T. and Pelánek, R. 2021a. Interpretable clustering of students' solutions in introductory programming. In *Proceedings of Artificial Intelligence in Education*, I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, and V. Dimitrova, Eds. Springer, 101–112.

Effenberger, T. and Pelánek, R. 2021b. Visualization of student-item interaction matrix. In *Visualizations and Dashboards for Learning Analytics*, M. Sahin and D. Ifenthaler, Eds. Springer, 439–456.

Emmott, A. F., Das, S., Dietterich, T., Fern, A., and Wong, W.-K. 2013. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD Workshop* *on Outlier Detection and Description*, L. Akoglu, E. Müller, and J. Vreeken, Eds. Association for Computing Machinery, 16–21.

FANCSALI, S. E., LI, H., AND RITTER, S. 2021. Toward scalable improvement of large content portfolios for adaptive instruction. In *Joint Proceedings of the Workshops at EDM*, T. W. Price and S. S. Pedro, Eds. Vol. 3051. CEUR Workshop Proceedings.

FANCSALI, S. E., LI, H., SANDBOTHE, M., AND RITTER, S. 2021. Targeting design-loop adaptivity. In *Proceedings of the 14th International Conference on Educational Data Mining*, S. I. Hsiao, S. Sahebi, F. Bouchet, and J. Vie, Eds. International Educational Data Mining Society, 323–330.

GOUTTE, C., LÉGER, S., AND DURAND, G. 2015. A probabilistic model for knowledge component naming. In *Proceedings of the 8th International Conference on Educational Data Mining*, O. C. Santos, J. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, M. C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, and M. C. Desmarais, Eds. International Educational Data Mining Society, 608–609.

GUPTA, N., ESWARAN, D., SHAH, N., AKOGLU, L., AND FALOUTSOS, C. 2019. Beyond outlier detection: LookOut for pictorial explanation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds. Springer, 122–138.

HASSANI, H., SILVA, E. S., UNGER, S., TAJMAZINANI, M., AND MAC FEELY, S. 2020. Artificial intelligence (ai) or intelligence augmentation (ia): What is the future? *AI 1,* 2, 143–155.

HUANG, Y., LOBCZOWSKI, N. G., RICHEY, J. E., MCLAUGHLIN, E. A., ASHER, M. W., HARACKIEWICZ, J. M., ALEVEN, V., AND KOEDINGER, K. R. 2021. A general multi-method approach to data-driven redesign of tutoring systems. In *Proceedings of the 11th International Learning Analytics and Knowledge Conference*, M. Scheffel, N. Dowell, S. Joksimovic, and G. Siemens, Eds. Association for Computing Machinery, 161–172.

KAVITHA, R., VIJAYA, A., AND SARASWATHI, D. 2012. Intelligent item assigning for classified learners in ITS using item response theory and point biserial correlation. In *Proceedings of 2012 International Conference on Computer Communication and Informatics*, S. Zhong, Ed. IEEE, 1–5.

KHOSRAVI, H., SHUM, S. B., CHEN, G., CONATI, C., TSAI, Y.-S., KAY, J., KNIGHT, S., MARTINEZ-MALDONADO, R., SADIQ, S., AND GAŠEVIĆ, D. 2022. Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence 3*, 100074.

KIRILENKO, A. P., STEPCHENKOVA, S. O., AND DAI, X. 2021. Automated topic modeling of tourist reviews: does the anna karenina principle apply? *Tourism Management 83*, 104241.

KOEDINGER, K. R. AND MCLAUGHLIN, E. A. 2016. Closing the loop with quantitative cognitive task analysis. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. International Educational Data Mining Society, 412–417.

KRATHWOHL, D. R. 2002. A revision of bloom's taxonomy: An overview. *Theory into practice 41,* 4, 212–218.

LINARDATOS, P., PAPASTEFANOPOULOS, V., AND KOTSIANTIS, S. 2020. Explainable AI: A review of machine learning interpretability methods. *Entropy 23,* 1, 18.

LIU, B., XIA, Y., AND YU, P. S. 2000. Clustering through decision tree construction. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, A. Agah, J. Callan, E. Rundensteiner, and S. Gauch, Eds. Association for Computing Machinery, 20–29.

LIU, F. T., TING, K. M., AND ZHOU, Z.-H. 2012. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD) 6,* 1, 1–39.

LIU, R. AND KOEDINGER, K. R. 2017. Closing the loop: Automated data-driven cognitive model discoveries lead to improved instruction and learning gains. *Journal of Educational Data Mining 9,* 1, 25–41.

MIAN, S., GOSWAMI, M., AND MOSTOW, J. 2019. What's most broken? design and evaluation of a tool to guide improvement of an intelligent tutor. In *Proceedings of Artificial Intelligence in Education*, S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, Eds. Springer, 283–295.

MOHAMMED, M. AND OMAR, N. 2020. Question classification based on bloom's taxonomy cognitive domain using modified tf-idf and word2vec. *PloS one 15,* 3, e0230442.

OMAR, N., HARIS, S. S., HASSAN, R., ARSHAD, H., RAHMAT, M., ZAINAL, N. F. A., AND ZULKIFLI, R. 2012. Automated analysis of exam questions according to bloom's taxonomy. *Procedia-Social and Behavioral Sciences 59*, 297–303.

OOSTERHOF, A. C. 1976. Similarity of various item discrimination indices. *Journal of Educational Measurement 13,* 2, 145–150.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

PELÁNEK, R. 2016. Applications of the elo rating system in adaptive educational systems. *Computers & Education 98*, 169–179.

PELÁNEK, R. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction 27,* 3, 313–350.

PELÁNEK, R. 2018. The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction 28,* 3, 207–235.

PELÁNEK, R. 2020a. Managing items and knowledge components: domain modeling in practice. *Educational Technology Research and Development 68,* 1, 529–550.

PELÁNEK, R. 2020b. Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies 13,* 2, 354–366.

PELÁNEK, R. 2021. Analyzing and visualizing learning data: A system designer's perspective. *Journal of Learning Analytics 8,* 2, 93–104.

PELÁNEK, R. 2022. Adaptive, intelligent, and personalized: Navigating the terminological maze behind educational technology. *International Journal of Artificial Intelligence in Education 32*, 151–173.

PELÁNEK, R. AND EFFENBERGER, T. 2022. Improving learning environments: Avoiding stupidity perspective. *IEEE Transactions on Learning Technologies 15,* 1, 64–77.

PELÁNEK, R., EFFENBERGER, T., AND ČECHÁK, J. 2022. Complexity and difficulty of items in learning systems. *International Journal of Artificial Intelligence in Education 32*, 196–232.

PELÁNEK, R. AND ŘIHÁK, J. 2018. Analysis and design of mastery learning criteria. *New Review of Hypermedia and Multimedia 24*, 133–159.

PELÁNEK, R., RIHÁK, J., ET AL. 2016. Properties and applications of wrong answers in online educational systems. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. International Educational Data Mining Society, 466–471.

PYRCZAK, F. 1973. Validity of the discrimination index as a measure of item quality 1. *Journal of Educational Measurement 10,* 3, 227–231.

RACHATASUMRIT, N. AND KOEDINGER, K. R. 2021. Toward improving student model estimates through assistance scores in principle and in practice. In *Proceedings of The 14th International Conference on Educational Data Mining*, S. I. Hsiao, S. Sahebi, F. Bouchet, and J. Vie, Eds. International Educational Data Mining Society, 295–301.

SAISUBRAMANIAN, S., GALHOTRA, S., AND ZILBERSTEIN, S. 2020. Balancing the tradeoff between clustering value and interpretability. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, A. Markham, J. Powles, T. Walsh, and A. L. Washington, Eds. Association for Computing Machinery, 351–357.

SALO, O. AND ABRAHAMSSON, P. 2007. An iterative improvement process for agile software development. *Software Process: Improvement and Practice 12,* 1, 81–100.

SHEEHAN, K. M., KOSTIN, I., NAPOLITANO, D., AND FLOR, M. 2014. The textevaluator tool: Helping teachers and test developers select texts for use in instruction and assessment. *The Elementary School Journal 115,* 2, 184–209.

STEINKE, A., KOPP, B., AND LANGE, F. 2021. The wisconsin card sorting test: split-half reliability estimates for a self-administered computerized variant. *Brain Sciences 11,* 5, 529.

WANG, X., ROSE, C., AND KOEDINGER, K. 2021. Seeing beyond expert blind spots: Online learning design for scale and quality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, K. Isbister, T. Igarashi, P. Bjørn, and S. Drucker, Eds. Association for Computing Machinery, 1–14.

YAHYA, A. A., TOUKAL, Z., AND OSMAN, A. 2012. Bloom's taxonomy–based classification for item bank questions using support vector machines. In *Modern advances in intelligent systems and tools*, W. Ding, H. Jiang, M. Ali, and M. Li, Eds. Springer, 135–140.

ZANZOTTO, F. M. 2019. Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research 64*, 243–252.