Data-driven cognitive skills with an application in personalized education

Niels Anne Taatgen (n.a.taatgen@rug.nl)
Corné Hoekstra (hoekstracorne@gmail.com)
Bernoulli Institute, University of Groningen, Nijenborgh 9
9747 AG Groningen, Netherlands

Jori Blankestijn (jori@memorylab.nl)
MemoryLab, Stationsstraat 10
9711 AS Groningen, Netherlands

Abstract

How can we explain that people are capable of performing new tasks with no or little instruction? Earlier work has proposed that new tasks can be acquired by a rapid composition of cognitive skills, and implemented this in the ACT-R and PRIMs cognitive architectures. Here, we discuss a possible application of rapid composition in building tutoring systems. The goal is to identify underlying skills through unsupervised machine learning from a dataset of arithmetic learning for students in a Dutch vocational program. The resulting skill graph is used as a basis for a tutoring system. The results show evidence for predictive power of the system and tentative evidence of a learning benefit compared to control groups.

Keywords: cognitive architecture; cognitive skills; tutoring systems; ACT-R

Introduction

Humans have the remarkable ability to carry out arbitrary new tasks if they have the right prior knowledge and skills. In psychological experiments, subjects can perform new tasks they have never done before with only a short instruction and a few practice trials, if any. Researchers hardly ever pay attention to this translation of a short instruction to a task representation (with exceptions, e.g., Cole, Bagic, Kass, & Schneider, 2010). Similarly, cognitive modelers tend to assume that the sometimes extensive representation of the task knowledge is somehow encoded in memory without explaining how. In reality, we have to assume that subjects base their task representation on a combination of elements of knowledge they already have. Earlier work by Anderson, Taatgen and Salvucci (Anderson et al., 2004; Taatgen, Huss, Dickison, & Anderson, 2008; Salvucci, 2013) has looked at *production rules* as a unit of reusable task knowledge within the context of the ACT-R cognitive architecture. However, production rules are relatively fine-grained units of representation, and many of them are needed even for simple tasks. Earlier, Card, Moran, and Newell (1983) identified the *unit task* as a basic unit of representation, without being clear on how these can be combined. Hoekstra, Martens, and Taatgen (2020) propose a level of representation that combines several production rules in larger, reusable units that they call *skills*, and show an implementation of these skills in the PRIMs cognitive architecture, an architecture derived from ACT-R (Taatgen, 2013). As a demonstration, they created models of a visual search task, a simple working memory task and a complex working memory task using skills. They then showed that without any further additions, a recombination of the skills from these tasks

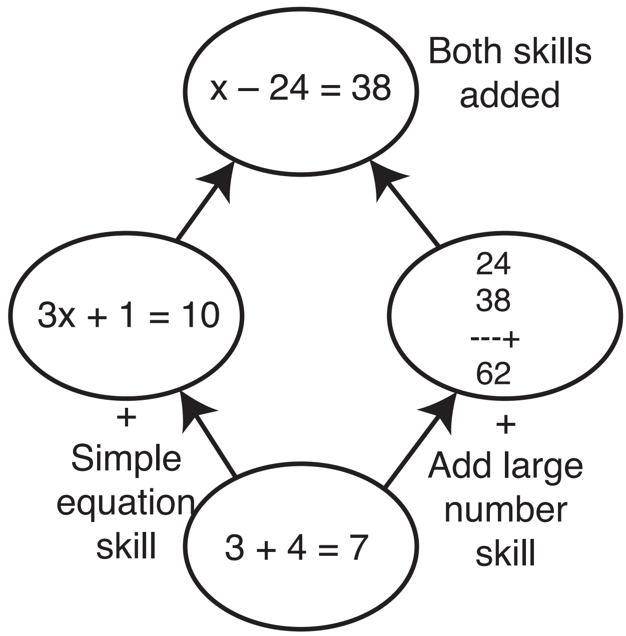

Figure 1: Example of a constructed knowledge graph

could model the Attentional Blink task (Raymond, Shapiro, & Arnell, 1992). In that task, subjects have to recognize two targets in a rapid stream of stimuli. If the stimuli are between 200 and 400ms apart, the second target is often missed: the attentional blink effect. Hoekstra et al. showed that the presence or absence of an attentional blink can be attributed to the choice of working memory skill.

Many research questions and challenges persist regarding the concept of higher-level skills as the foundation for cognitive modeling. One of the challenges is to identify which skills people have, and how individuals differ in their skills. In this paper, we pursue this challenge in an applied setting: personalized education in a classroom setting. If skills play a central role in our ability to flexibly carry out tasks, the acquisition of these skills should be a major topic of study. This goal is pursued by *cognitive tutors*, programs that maintain a target representation of the material to be learned, and the extent to which the student has already acquired this (Anderson, Corbett, Koedinger, & Pelletier, 1995). Target representations are hand-crafted by the modelers and developers. This

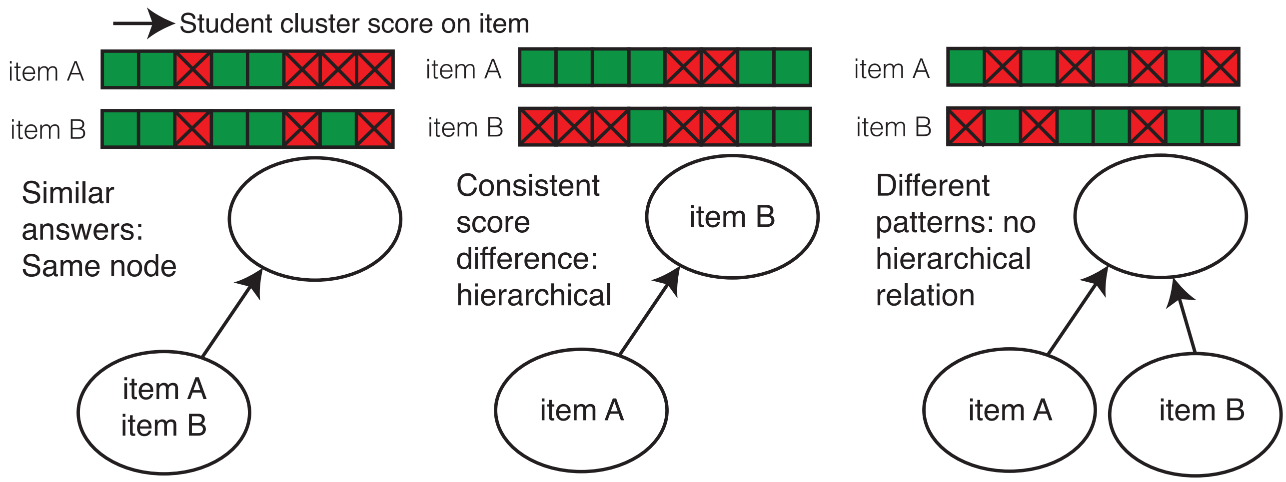

Figure 2: Illustration of how items are placed in the graph. Colored boxes represent student cluster scores on the two hypothetical items: green is correct and red is wrong. Left panel: if answer patterns are similar, items should be in the same node. Middle panel: if a correct answer on B implies a correct answer on A, there should be a hierarchical relation between the two. Right panel: if answer patterns are different, their should be no path between the nodes.

approach becomes a challenge when it is not entirely clear what the goal representations are, or how individuals differ in their prior knowledge. Even when the tutor is based on a clearly specified model, it forces the student to adhere to that particular model, disallowing potential alternative solutions (Waalkens, Aleven, & Taatgen, 2013).

An alternative to hand-crafting a tutoring system is to use data to derive the representation. This approach is used in Math Garden (Klinkenberg, Straatemeier, & van der Maas, 2011), in which study items (we will use the word "items" throughout to refer to problems or exercises that students need to solve) are ordered on the basis of the ease with which students solve the items. Math Garden sets up a competition between items and students, naturally sorting items with respect to difficulty. This works very well for items that differ in difficulty on a single dimension, for example multiplications, but once items become more diverse, Math Garden has to set up different competitions for different categories of items.

Instead of a linear ordering of all the items, they can be organized in a graph (Falmagne, Koppen, Villano, Doignon, & Johannesen, 1990). Each node in the graph can be associated with the presence or absence of particular skills. The bottom node in the graph then represents items that do not require any new skills (corresponding to the situation that all students already have the knowledge to solve these items), whereas to top node represents items that require all the skills. Intermediate nodes represent particular combinations of skills. For example, skills could be:

1. Arithmetic with large numbers
2. Simple equation solving

Figure 1 shows an example of a graph with some example items.

Constructed knowledge graphs, such as this one, have problems. First, we don't know whether they are appropriate for the target population. Perhaps not all students have mastered simple arithmetic, and we have to add it as an additional skill. Second, and more critical: how do we know that these are the skills that underly the material? It may be clear in this simple example, but if items are more complex, which they usually are, it may be hard to properly separate the relevant skills. Or, even worse, we have no clear idea what the underlying skills are, such as in learning programming.

We therefore propose an approach that is different from engineering the knowledge representations, and that is to look for patterns in data from students. The general idea is that if a group of students performs systematically better on a subset of items than another group, this is an indication that this subset requires one or more skills that the first group has, but the second group has not.

General Methodology

The input for the algorithm is an *item* $\times$ *student* matrix with item scores in the cells. The assumption of the method is that students who have very similar scores on items have similar skill sets. We therefore first run the k-means clustering algorithm (Lloyd, 1982) on the student dimension of the matrix. This reduces the matrix to an item x student cluster centroid matrix. This matrix is the starting point for building the graph. Building the graph involves assigning items to nodes. Generally, we want items on which all student clusters score similarly in the same node. But if part of the student clusters score better on item A than on item B, but the reverse is not the case, so very few student clusters score better on item B than on item A, item B has to be higher in the graph than item A, and there has to be path from A to B. On the other hand, if student clusters have very different answer patterns on items, they should not have a hierarchical relationship in the graph. Figure 2 illustrates these concepts.

The challenge is to find a solution that works for each pair of items. Generally, there will not be a perfect solution, but so we are looking for a solution that is closest to perfect. First, we have to define the size of the graph in terms of the number of skills. For example, our Figure 1 graph has only two skills, producing a 2^2-node graph. We then define penalties for each pair of items to the extent they violate the constraints illustrated in Figure 2. Finally, we use simulated annealing to minimise the total penalty. In particular we use the GenSA library in R (Xiang, Gubian, Suomela, & Hoeng, 2013).

To decide on the number of skills, we use the remaining summed penalty as a guideline. Each time we want to add a skill, we have to check whether this significantly reduces the penalty. If it doesn't, we have found the right number of skills that can be derived from the data.

Once the graph building algorithm has generated the graph, the nature of the skills has to be identified. This requires human interpretation by comparing items between the nodes, in particular by checking what items within a node have in common, and what skill is added when comparing nodes that have a direct arrow between them. This interpretation should ideally be carried out be education experts, but in this study it was performed by the researchers themselves.

Exploratory Study

To test whether the graph-based approach produces measurable advantages in an educational setting, we conducted a study in collaboration with the Alfa College in Groningen, and Noordhoff publishers. The target population are first-year students in the MBO (Middelbaar Beroepsonderwijs, Mid-level vocational education, ages 16-20). All Dutch MBO students have to take and pass an obligatory course in arithmetic. The challenge in teaching this course is that students come from very varied backgrounds, and therefore vary considerably in the knowledge and skills when starting the course.

Noordhoff publishers provides educational materials for this course, including an online environment in which students can take tests, and practice on example items. The students in the course used this environment for all their practice work. The course consists of five units, each of which are examined separately. This study focused on the first of these units. At the Alfa College, this first unit is taught in a period of ten weeks. In the first week, students take a pretest of 21 items that covers all the topics of the first unit. For the next seven weeks they cover the seven topics that make up unit 1 with two lessons per week. They then do a midterm exam as practice, followed by three more weeks of practice leading up to the final exam at the end.

Method

For this study, we used pretest results from the previous year to construct the knowledge graph, and to assign items to nodes. This data consisted of results from 2480 students from 2022.

Subjects Three groups of students from the Alfa College participated in the study, one experimental, and two groups served as control. The experimental group consisted of 24 students, and the two control groups consisted of 23 and 19 students.

Materials The Noordhoff method for arithmetic consists of a textbook and an online environment in which the students can make assignments, both the pretest and practice items for the seven topics. The topics were the following:

1. Numbers and units
2. Length
3. Weight
4. Time
5. Other measures
6. Reference measures
7. Rules of the thumb and equations

The pretest consists of 21 items. The course materials themselves consist of 106 items divided over the seven topics.

Procedure On the basis of the results from the previous year, a knowledge graph was constructed. Subsequently, on the basis of the analysis of the content of the nodes, items from the course materials were assigned to the nodes in the graph. Therefore, each node consisted of items from the pretest, and items from the topics of the lessons.

In the first week of the course, students took the pretest, producing an initial estimate on how well they performed on each of the nodes in the graph. This was presented to them in an interactive interface, in which they could see their performance on each node, and in which they could click on each node to select additional practice items belonging to that node.

In each of the subsequent seven lesson weeks, students used the interactive interface to select practice items that belonged to the topic of the week. The students were encouraged to select items at the level indicated by the graph, but were given freedom to select their own strategy. Twice per week the graph was updated to reflect the students' current mastery of each of the nodes.

After seven weeks, students took a practice (midterm) exam. After the midterm exam, they had three more weeks to use the interactive interface to select additional practice items. They were again given the freedom to pick any item they wanted, but were encouraged to pick items that were in nodes that they still needed more practice with. During this period, we gave them a survey, in which we asked them about their experience with the system.

At the end of the three weeks, they took a final exam. The two control classes followed the same curriculum as the experimental class, with the same topics in each of the weeks, and the same pretest and practice items, but without the interactive interface.

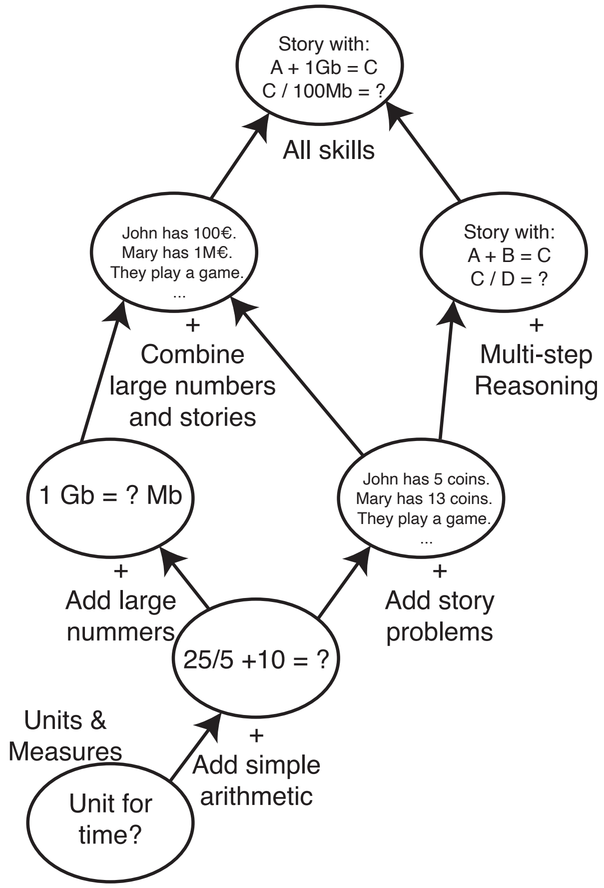

Figure 3: Knowledge graph from pretest for the first unit of the arithmetic course. Note that the example items in the nodes are not from the actual test: the real items are all quite elaborate and would not fit the figure. Each node represents a subset of the four skills, with the top node all four skills, and the bottom node no skills.

Graph Construction and Interface

Knowledge Graph

To determine the optimal number of student clusters, we used AIC as the criterion, This revealed that the optimal number in the data from the previous year was 11. We used those 11 clusters as the basis for the graph algorithm, and found that four skills provides the best description: there is hardly any gain in error with five skills. An examination of the items and related skills revealed that the best explanation for the individual differences is to assume the following four skills:

Simple arithmetic The ability to do simple calculations, including addition, subtraction, multiplication and division.

Large numbers and estimation To have a sense of numbers beyond small numbers, such as numbers with many zeros, calculations with kilo/mega/giga, and making estimations.

Story problems The ability to translate a verbal description into a calculation.

Multi-step reasoning The ability to carry out calculations that require multiple calculations steps.

Figure 3 shows the graph. Problems that involved only units and measurements were made correctly ($> 90\%$) by almost all students, so we can consider this a skill students already have. Basic arithmetic is the first skill that some students have not mastered, and is required before any of the others. Problems that require multi-step reasoning as a skill always also require the Story problem skill, because students have to deduce these steps from the text.

Interactive Interface

On the basis of the knowledge graph we constructed the interactive website depicted in Figure 4. The graph has small icons representing the skills involved, which are also listed on the right side of the screen. Students were able to see their performance on each of the nodes, and could click on nodes to see scores on individual items and items they had not yet tried. They could also click on the arrows on the top-left of the screen, allowing them step through the history of the developing graph. Clicking an item would take them to the Noordhoff website on which they could try to solve the item. The graph was updated twice per week (i.e. before each classroom session) to show the student's progress.

Results

Survey

Table 1 shows the results of the survey. Students were positive about the system and the insight it gave them. They also indicated that it did not improve their motivation for the course, and were uncommitted about the remaining questions.

Table 1: Survey results ($n = 12$). Scores are on a five point Likert scale.

Question	Score
Star rating (max 5 stars)	3.75
Did the program give you insight in your level of arithmetic?	3.92
Did the program motivate you for arithmetic?	2.58
Did you do more exercises?	3.08
Did you do more useful exercises?	3.25
Was practice with the system useful for the final test?	3
Was practice with the system useful for the individual topics?	3.17

In open questions students indicated they were generally positive about the systems, and found it easy to make decisions. They did not like the fact that they had to switch

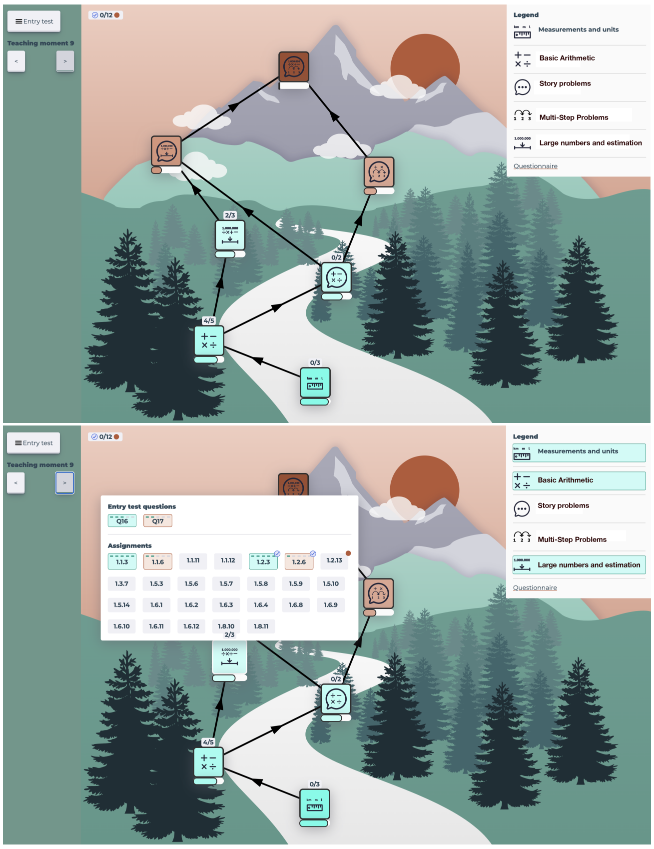

Figure 4: Interactive interface depicting the graph for an individual student. Nodes are color-coded to reflect the level of mastery (brown to green), and additionally show a progress bar. Students can click on or hover over a node (lower panel) to see the items they have attempted before, and the score they obtained. They can click on items they have not yet tried, which takes them to the Noordhoff website on which they do the item. Translated into English, original is in Dutch.

between two websites: the website with the graph, and the website with the assignments.

Predictive Power of Node Score

If the graph gives a good assessment of the state of the skills of an individual student, it should have predictive power on the probability of success for a new item for that node. To test this, we fitted a linear mixed effects model to predict the score on new items. We also added student ID as a random effect, to correct for individual differences. Data from both the experimental and the control groups were used, because the predictive power can be tested even if the students themselves did not use the system.

The node score ($\beta = 0.15, \text{std} = 0.025, \text{df} = 1934, t = 5.966, p < 0.001$) has a significant impact on the score. Figure 5 illustrates the predictive power of node score. The graph shows the student score on a node on the x axis, and the deviation from the average score of that student for new items in

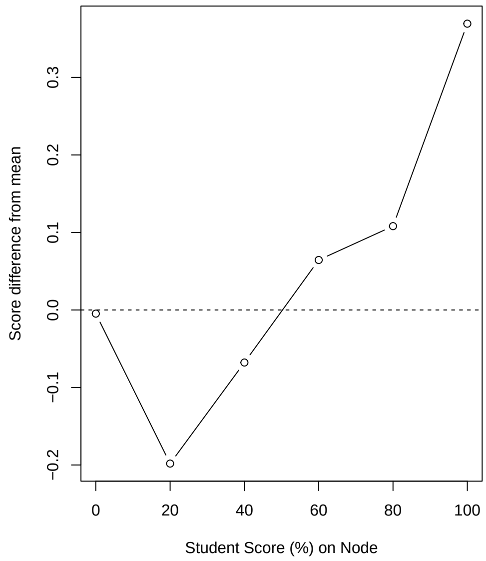

Figure 5: Predictive power of the system

that node on the y axis.

Comparison between Experimental and Control Groups

It is not possible to make a reliable comparison between the experimental and control groups, because they are taught by different teachers. In addition, not all students make all the tests, and students drop out of the course. We therefore chose not to perform a statistical test on the data, because this would suggest an exactness that we cannot warrant.

Nevertheless, it is interesting to look at the results from the students. Figure 6 shows the average scores on the pretest, the midterm test, and the final test. Note that the tests do not have the same difficulty, so we cannot conclude that students have learned nothing. However, we do not see any difference between the experimental group and the two control groups with respect to test score.

The scores do not tell the whole story. Students frequently drop out of the course. Figure 7 presents a different view on how well students are doing: here we see that in the Control groups more students dropped out, and that in the end the passing rate for the Experimental group is 58%, compared to 35% and 37% in the Control groups.

Discussion

The skills identified in the graph approach are quite different from the learning goals set by the educational publisher. This does not mean that those goals are wrong, but rather that we look at a different aspect of learning. Instead, both can be considered orthogonal. The learning goals are focused on the specific topics and knowledge, but the skills are more general. Also, the publisher's learning goals concern topics

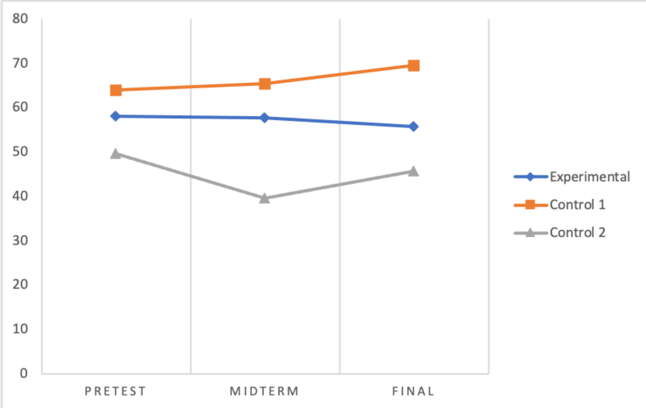

Figure 6: Test scores on pretest, midterm and final test. Scores are on scale between 0 and 100.

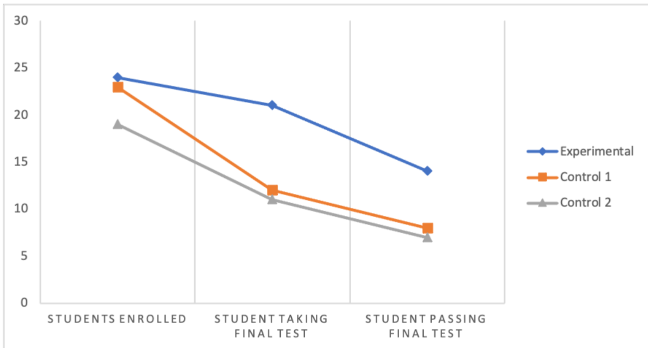

Figure 7: Enrollment, students taking the final test, and students passing the final test.

that all students have been exposed to before in elementary school. This means there is no or very little really new material. For some of the students, though, the skills to handle assignments on these topics may be absent, or have been forgotten. We therefore believe the approach of going through the topics following the course materials, but with an individual focus on weak skills may be the optimal way of offering the material.

In the results we see some appreciation of the students for the system, even though it did not increase their motivation for the topic. They did not like switching back and forth between websites. For this experiment, integration of the two websites was not feasible, but in a future application this is recommended.

The data show that the assignment of items to nodes has predictive power, and can therefore help students select the right item to study. We could demonstrate this on the dataset as a whole, so also for the control groups. Therefore it is possible to conclude that working with the order proposed in the system's graph provides opportunities for students to identify areas of optimal difficulty for them (zone of proximal development). Also, choosing their own path instead of using a predetermined path teaches autonomy and autonomy in choosing exercises has been shown to improve learning results.

The comparison between the groups suggested that the passing rate of the experimental group is quite a bit higher, because fewer students drop out of the course. However, we have to be careful with this conclusion, because this may be attributable to other factors, even though we did not identify any. A larger study is needed to substantiate this result.

The approach in this paper has been inspired by theoretical work in cognitive architectures. The grain-size of the skills identified from the data is not as fine as that in the theoretical models, as we already saw in work by Akrum and Taatgen (2023). Bridging this gap requires more fine-grained data, potentially with items that are designed to find subtle distinctions. It also depends on the data: if students do not differ on the dimension of a particular skill, the methodology presented here will not find it.

Our approach has parallels with *Knowledge Space Theory* (Falmagne et al., 1990; Doignon & Falmagne, 2012). The perspective from the viewpoint of cognitive architectures and skills is different from that theory, as well as the algorithm to construct the graph, but it is worthwhile to explore synergy between the approaches.

Limitations

This study has a number of limitations. As indicated, the experimental group only consisted of a single group of students. The difference in outcome between that group and the two control groups might be attributable to other, unknown factors. Another limitation is that the skills that have been assigned to nodes in the graph have been identified by the authors. Subsequently, additional test items were assigned to the nodes on the basis of this identification. There are several future options to mitigate this, by asking multiple education experts to identify skills, or to use Large Language Models for the identification.

Conclusions

In this paper, we have shown that the theory of skills can be used as a basis to construct skill graphs on the basis of student data. An experimental study to test the educational benefits of the approach shows promising, but not definite results.

Acknowledgments

This project was funded by the ECP platform voor de Informatiesamenleving. We also want to thank the Alfa College Groningen, and Siemen Bleijenberg for their help.

References

Akrum, I. D. M., & Taatgen, N. A. (2023). From knowledge graph to cognitive model: A method for identifying task skills. In *Proceedings of iccm-2023.* Retrieved from `https://mathpsych.org/presentation/1057/`

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*, 1036–1060.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: lesson learned. *The Journal of the Learning Sciences*, *2*(2), 167–207.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Cole, M. W., Bagic, A., Kass, R., & Schneider, W. (2010). Prefrontal Dynamics Underlying Rapid Instructed Task Learning Reverse with Practice. *Journal of Neuroscience*, *30*(42), 14245–14254.

Doignon, J.-P., & Falmagne, J.-C. (2012). *Knowledge spaces*. Springer Science & Business Media.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, *97*(2), 201.

Hoekstra, C., Martens, S., & Taatgen, N. A. (2020). A skill-based approach to modeling the attentional blink. *Topics in Cognitive Science*, *12*(3), 1030–1045.

Klinkenberg, S., Straatemeier, M., & van der Maas, H. (2011). Computer adaptive practice of Maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, *57*(2), 1813–1824. doi: 10.1016/j.compedu.2011.02.003

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, *28*(2), 129–137.

Raymond, J. E., Shapiro, K. L., & Arnell, K. M. (1992). Temporary suppression of visual processing in a RSVP task: An attentional blink? *Journal of Experimental Psychology: Human Perception and Performance*, *18*, 849–860.

Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive science*, *37*(5), 829–60. doi: 10.1111/cogs.12032

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*, 439–471.

Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008, aug). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General*, *137*(3), 548–565.

Waalkens, M., Aleven, V., & Taatgen, N. (2013). Does supporting multiple student strategies lead to greater learning and motivation? investigating a source of complexity in the architecture of intelligent tutoring systems. *Computers & Education*, *60*(1), 159–171.

Xiang, Y., Gubian, S., Suomela, B., & Hoeng, J. (2013). Generalized simulated annealing for efficient globaloptimization: the gensa package for r. *The R Journal*, *5/1*. Retrieved from `https://journal.r-project.org`