Measuring the Quality of Domain Models Extracted from Textbooks with Learning Curves Analysis

Isaac Alpizar-Chacon[1,2(✉)], Sergey Sosnovsky[1], and Peter Brusilovsky[3]

[1] Utrecht University, Utrecht, The Netherlands
{i.alpizarchacon,s.a.sosnovsky}@uu.nl
[2] Instituto Tecnológico de Costa Rica, ATI, Compus Cartago, Costa Rica
ialpizar@itcr.ac.cr
[3] University of Pittsburgh, Pittsburgh, USA
peterb@pitt.edu

Abstract. This paper evaluates an automatically extracted domain model from textbooks and applies learning curve analysis to assess its ability to represent students' knowledge and learning. Results show that extracted concepts are meaningful knowledge components with varying granularity, depending on textbook authors' perspectives. The evaluation demonstrates the acceptable quality of the extracted domain model in knowledge modeling.

Keywords: Knowledge Extraction · Learning Curves · Textbooks

1 Introduction

Automation of the creation of domain models (DMs) has been a long-standing practical and research problem in the field of Artificial Intelligence in Education. Most types of adaptive educational systems require high-quality composite fine-grained representations of domain knowledge to be able to model students' abilities and provide meaningful support of their learning. Development of such representations has been traditionally a manual task demanding a great deal of time and expertise. Over the last decade, a range of approaches have been introduced capable of extracting different elements of domain semantics from domain-oriented documents and user data [6,7,11]. However, very few studies examined the applicability of automatically extracted domain semantics to the task of modeling student knowledge [13].

This paper presents an evaluation of a DM that has been automatically extracted from a collection of textbooks in the same domain—Python programming. Section 2 briefly outlines the approach developed to produce such a model. The quality of the DMs extracted from textbooks with this approach has been evaluated before in terms of accuracy [2], semantic completeness [1], coverage [1], and domain specificity [4]. This paper explores the quality of the concepts

N. Wang et al. (Eds.): AIED 2023, LNAI 13916, pp. 804–809, 2023.
https://doi.org/10.1007/978-3-031-36272-9_75

extracted from textbooks as Knowledge Components (KCs) assessing their cognitive validity and applicability for knowledge modeling and assessment. Additionally, the concepts are evaluated to see if they cover too much or too little knowledge (granularity).

The best approach to validate the extracted concepts from this perspective is the *learning curve analysis* [9][1]. Learning curves are graphs that plot performance on a task versus the number of attempts to practice. Performance is usually measured using the proportion of incorrect responses (the error rate) for a KC that is being practiced. Learning curve analysis is used to qualify learning performance. If learning occurs for the KC being measured, the learning curve should follow the power law [10]. That is, the error rate of a KC should decrease as a power function of the number of attempts involving this component. A positive slope (α) indicates a decreasing curve and, therefore, a learning effect. A high fit (R^2) indicates that the KC successfully identifies the student's learning.

Motivated by the mentioned learning curve analysis, this paper describes an experiment to assert the cognitive validity and granularity of concepts extracted from textbooks. The analysis of the learning curves showed the DMs automatically extracted from textbooks consist of cognitively valid knowledge components for domain knowledge modeling. Additionally, textbooks provide both fine- and coarse-grained concepts; smaller concepts are shown to support more accurate student modeling.

2 Background

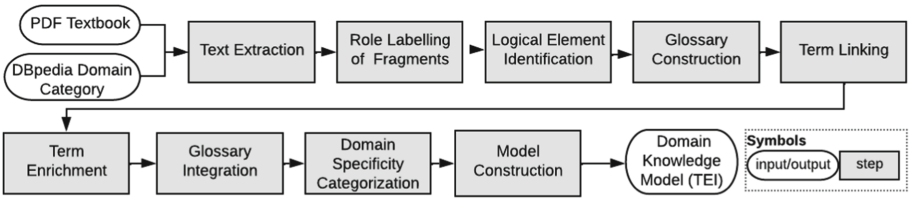

Fig. 1. Stages for the extraction of DMs from textbooks.

We have developed a workflow for the automated extraction of DMs from textbooks [1–4]. Figure 1 shows the main stages of this approach. The first three stages use an extensive set of rules that capture common conventions and guidelines for textbook formatting, structuring, and organization. The textbook's structure (chapters and subchapters), content (words, lines, text fragments, pages, and sections), and domain terms (terminology used in the textbook and the domain) are extracted. In the following three stages, the domain terms are used as a bridge to link the textbooks to entities in DBpedia[2]. The linking with

[1] The reader is directed to this source for a comprehensive introduction to learning curves.

[2] https://www.dbpedia.org/.

DBpedia allows for the enrichment of the domain terms with semantic information (e.g., abstracts and categories). In the seventh stage, terms from multiple textbooks are integrated into a single model to get better coverage of the target domain. Then, in the next stage, terms are categorized according to their relevance to the target domain (main, related, or unrelated domain). Finally, all the extracted knowledge is serialized as a descriptive XML file[3].

3 Experiment

This experiment examines the conceptual representation of knowledge in DMs from textbooks, exploring the validity of concepts as cognitive KCs and analyzing their granularity. Learning curve analysis is used to quantify learning performance for these concepts.

Data. Concepts extracted from three introductory Python programming textbooks[4] are analyzed using learning activities and learner data from PythonGrids [5], a personalized practice system for Python programming. PythonGrids learning activities are grouped into 15 ordered topics, from simple (e.g., "Variables and Operations") to advanced (e.g., "Classes/Objects"). Eleven datasets of students' interactions with PythonGrids are used for this experiment[5]. A final combined dataset containing 57929 interactions of 465 students with 85 activities was used in this experiment.

Procedure. The experiment has four steps. The first step is extracting the domain knowledge from textbooks and selecting the relevant concepts. Then, the learning activities are annotated with the selected concepts according to their expected learning outcomes. After that, the interactions from the students are aggregated, filtered out, and augmented. Finally, learning curves are generated for each concept.

1. Domain Knowledge Generation. A DM is extracted and enriched for each textbook (see Sect. 2). The models are combined into a single model to merge repeated terms. The model contains 600 terms, with 266 considered relevant for Python programming, which are used to annotate learning activities.

2. Content Annotation. Learning activities are annotated with the selected concepts by experts to indicate expected learning outcomes. Annotators assess topics, prerequisite/outcome relations, and activity outputs to choose the relevant concepts. In total, 54 concepts are used in annotations (the KCs), including "variable", "function call", "for loop", "function", and "exception".

[3] https://tei-c.org/.
[4] Python for everybody, Think Python, and Introduction to computation and programming using Python.
[5] Provided through the PSLC DataShop at http://pslcdatashop.web.cmu.edu.

3. Data Preparation. The used dataset comes from students using the PythonGrids system in real and diverse settings with no control over the environment. Therefore, the data have to be treated with caution. Reliable learning sequences that help to evaluate the concepts need to be extracted from the data. Sequences with no evidence of learning have to be regarded as noise. First, the interactions in the dataset are grouped into sequences containing all student attempts per concept [12]. In total, 10946 student-concept-attempts are generated. Correct outcomes are marked with 1's, incorrect ones with 0's. After that, we identify rapid-guessing [8] to reduce the number of incorrect attempts (smoothing) in the sequences of activities where the trial-and-error strategy works exceptionally well. Then, noise in the student-concept-attempts sequences is filtered out. Sequences are labeled with four tags: *known*, *understood_strong*, *understood_weak*, and *not_understood*. These tags identify students who already know the concept, have practiced until mastery, practiced until getting a correct answer, or stopped before showing any learning, respectively. Sequences without learning (*known* and *not_understood*) are filtered out, leaving 8079 student-concept-attempts (73.8% of all sequences). After filtering, new attempts are generated by augmenting student-concept-attempts sequences to maintain the learning evidence for students who stopped practicing earlier. This ensures the same number of concept-attempts for each student. Correct attempts (1's) are inserted in *understood_strong* sequences, while the average of the original attempts are inserted in *understood_weak* sequences until the maximum number of attempts is reached.

4. Learning Curves Generation. The learning curves are generated using the processed student-concept-attempts sequences. For each concept, the error rates at each attempt are calculated using $1 - \frac{sum\ of\ all\ outcomes}{total\ number\ of\ outcomes}$. After generating and analyzing the learning curves, a cut-off point was selected when the number of attempts was less than 25% of the first attempts. This threshold maintains a good balance between the number of attempts and the fit of the learning curves.

4 Results and Analysis

In total, 46 unique learning curves have been generated (eight concepts have had learning curves identical to other concepts'). Figure 2 displays representative examples of learning curves to guide the discussion of the results. All 46 learning curves are available online[6].

Cognitive Validity. Five[7] of 46 concepts show no learning, while the remaining 41 display a positive learning trend. The mean fit (R^2) of 0.65 ($SD = 0.27$) for these 41 positive curves indicates that **the assessed textbook concepts are cognitively valid units of knowledge**. The results align with similar literature evaluating fine-grained knowledge components [9,12]. Learning curves are classified using the power law parameters, with higher fit indicating more

[6] https://github.com/isaacalpizar/learning_curves.

[7] "Exception", "conditional statement", "value", "variable", and "iteration".

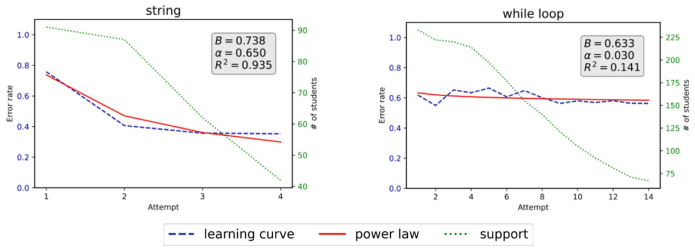

Fig. 2. Examples of learning curves.

reliable learning and steeper slope suggesting faster learning. Of the 41 downward learning curves, four are high-quality, 28 are medium-quality, and nine are low-quality. High-quality curves, such as "string" ($R^2 = 0.94$, $\alpha = 0.65$, Fig. 2-left), demonstrate more and faster learning compared to the other categories, with other concepts in this category including "hello, world", "optional parameter", and "counter". A low-quality learning curve corresponds to the "while loop" concept (Fig. 2-right), with a fit ($R^2 = 0.14$) indicating some learning but a low slope ($\alpha = 0.03$) suggesting slow progress. This may imply students struggle with while loops and that activities could be improved. Concepts "instance" and "reference" have similar curves.

Granularity. Regarding granularity, concepts in only one topic in PythonGrids are considered fine-grained (e.g., "float" in "Variables and Operations"), while those in multiple topics are coarse-grained (e.g., "iteration" in "While Loops" and "For Loops"). Fine-grained concepts have acceptable learning curves, while coarse-grained concepts are a special case. Out of these concepts, all but "iteration" produce downward learning curves, though not smooth. We can analyze the "indentation" concept. In Python, indentation is semantically meaningful and is used to indicate a block of code in many statements or expressions. This concept is linked to the "conditional statement", "while loop", "for loop", and "function" concepts. The learning curve for "indentation" is uneven with multiple upticks (bumps). The other coarse-grained concepts show similar learning curves. "Bumpy" learning curves may result from concepts being learned in the context of associated concepts. When a new associated concept is introduced, the probability of making a mistake is high, but the central concept eventually shows a downward trend after all associated concepts have been trained. This finding aligns with previous studies observing worse learning curves for more general groupings [9,12].

5 Conclusion and Future Work

This paper explored the quality of DMs extracted from textbooks regarding their ability to model students' knowledge and learning. Learning curve analysis showed textbook concepts in *Python programming* measure students' learning

(cognitive validity) and displayed different granularity levels. In conclusion, this paper provided strong evidence of the richness of the extracted models for domain modeling and assessment. Future work includes a more complex experiment with an educational system designed with textbook concepts at the core.

References

1. Alpizar-Chacon, I., Sosnovsky, S.: Expanding the web of knowledge: one textbook at a time. In: Proceedings of the 30th on Hypertext and Social Media, HT 2019, ACM, New York, NY, USA (2019)
2. Alpizar-Chacon, I., Sosnovsky, S.: Order out of chaos: construction of knowledge models from pdf textbooks. In: Proceedings of the ACM Symposium on Document Engineering 2020, pp. 1–10 (2020)
3. Alpizar-Chacon, I., Sosnovsky, S.: Knowledge models from pdf textbooks. New Rev. Hypermed. Multimed. **27**(1–2), 128–176 (2021)
4. Alpizar-Chacon, I., Sosnovsky, S.: What's in an index: extracting domain-specific knowledge graphs from textbooks. In: 2022 Proceedings of the ACM Web Conference (WWW 2022) (2022)
5. Brusilovsky, P., et al.: An integrated practice system for learning programming in python: design and evaluation. Res. Pract. Technol. Enhanced Learn. **13**(18), 1–40 (2018)
6. Chaplot, D.S., Yang, Y., Carbonell, J., Koedinger, K.R.: Data-driven automated induction of prerequisite structure graphs. Int. Educ. Data Min. Soc. (2016)
7. Chau, H., Labutov, I., Thaker, K., He, D., Brusilovsky, P.: Automatic concept extraction for domain and student modeling in adaptive textbooks. Int. J. Artif. Intell. Educ. **31**(4), 820–846 (2021)
8. Kong, X.J., Wise, S.L., Bhola, D.S.: Setting the response time threshold parameter to differentiate solution behavior from rapid-guessing behavior. Educ. Psychol. Measur. **67**(4), 606–619 (2007)
9. Martin, B., Mitrovic, A., Koedinger, K.R., Mathan, S.: Evaluating and improving adaptive educational systems with learning curves. User Model. User-Adapt. Interact. **21**(3), 249–283 (2011). https://doi.org/10.1007/s11257-010-9084-2
10. Newell, A., Rosenbloom, P.: Mechanisms of skill acquisition. Cognitive skills and their acquisition (1981)
11. Pan, L., Li, C., Li, J., Tang, J.: Prerequisite relation learning for concepts in MOOCs. In: The 55th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (2017)
12. Sosnovsky, S., Brusilovsky, P.: Evaluation of topic-based adaptation and student modeling in QuizGuide. User Model. User-Adapt. Interact. **25**(4), 371–424 (2015)
13. Wang, M., Chau, H., Thaker, K., Brusilovsky, P., He, D.: Knowledge annotation for intelligent textbooks. Technol. Knowl. Learn. **28**, 1–22 (2021). https://doi.org/10.1007/s10758-021-09544-z