Exploring the Content Ecosystem of the First Open-source Adaptive Tutor and its Applications on Intelligent Textbooks

Ioannis Anastasopoulos [1]

[1] *UC Berkeley, Berkeley, California, United States of America*

Abstract

The content ecosystem is an often overlooked component of adaptive tutors and intelligent tutoring systems. A robust and scalable content environment and creation process is critical in sustaining the long-term usage and subject area expansion of a tutor. Open Adaptive Tutor (OATutor), a recently introduced open-source adaptive tutor, produced three textbooks worth of creative commons content with custom built hints and scaffolds and problems sourced from OpenStax. In this paper, we examine how OATutor facilitated the creation of its content over a three-year period, as well as explore its content ecosystem, including the creation process and content team structure. We argue that the created content falls into the realm of intelligent textbooks and can be incorporated into digital textbooks to make them interactive.

Keywords

Intelligent Tutoring Systems, Crowdsourcing, Intelligent Textbooks, Creative Commons, Learnersourcing

1. Introduction

Intelligent Tutoring Systems (ITS) have consistently been shown to produce significant learning gains over the past thirty years serving as an indelible paradigm for computer supported instruction based on mastery learning [2]. Any adaptive system cannot succeed in achieving such goals without a robust content environment. The content ecosystem of a tutor plays a critical role in facilitating its usage and adoption, with a substantial portion of early ITS research spent on successfully lowering time and costs necessary to author content [3].

Until recently, there has been no tutor based on ITS principles that has made its content openly available via a creative commons or other permissive license. A new open-source tutoring system, OATutor [1], has created three algebra texts worth of material, adapted from OpenStax textbooks, and made it available under a creative commons BY license. To accomplish this, OATutor established a content environment inspired by the rapidly developing ideas of learnersourcing, allowing for replicable content creation and curation cycles in addition to its creative commons licensed content [1].

In this paper, we will discuss how the content environment and ecosystem of OATutor compares to proceeding tutors, how implementations of learnersourcing-inspired practices helped the OATutor project facilitate a replicable and scalable content library and content creation routine, and applications of such an ecosystem in the realm of intelligent textbooks.

2. Related Work

2.1. Content Creation Environments

For the purpose of enabling users to create content, adaptive tutors and ITS have historically featured their own unique content creation environments. The earliest content authoring tools for ITS required 200-300 hours of development for the production of a single instructional hour's worth of content [4].

Fifth International Workshop on Intelligent Textbooks (2023), July 03, 2023, Tokyo, Japan
EMAIL: ioannisa@berkeley.edu (A. 1)
ORCID: 0000-0002-1341-5876 (A. 1)

Such time requirements were quickly reduced, as Carnegie Learning developed the Cognitive Tutor Authoring Tools (CTAT) allowing the development time to drop to 50-100 hours for an instructional hour's worth of content [5]. CTAT provided a graphical user interface (GUI) that allowed for creation of user content with minimal knowledge of coding and programming skills. Content creation could take place directly on the interface without needing interference or assistance from external sources, allowing for many independent content creation processes [4].

With the success of CTAT, many emerging adaptive tutors facilitated their content creation through a set of builder tools focused around providing content creators with an effective GUI. A prime example of such tutors is ASSISTments, and the ASSISTments Builder functionality. ASSISTments is an adaptive tutor designed with the instructor as the focus [6]. Reflecting that design philosophy, the ASSISTments Builder was created with the intent to simplify content creation for instructors [3]. The builder provides content creators with a GUI designed for easy implementation of student help in the form of hints and scaffolding, in addition to skill mapping of knowledge components. Additionally, it successfully matches the lower end of CTAT's estimate of 50 hours of development for an instructional hour's worth of content while also eliminating any requirement for external programming knowledge [7]. The ASSISTments Builder also introduced support for variabilization directly into its GUI content builder, thus helping facilitate additional potential for a content creator to easily generalize their problems [3]. Overall, these features became standard in many future GUI problem builders, and established a baseline for what features a content builder tool should be able to support.

While GUI builders have increased in popularity due to the aforementioned innovations, there are various challenges still present in their development. It can be difficult for instructors to constantly have to learn new interfaces for software to incorporate into their curricula [8]. New systems are rapidly developed and introduced to the educational environment, and with content builders and interfaces being unique, it can be challenging to onboard individuals and have them dedicate the necessary time to familiarize themselves with the builder. Once instructors get past the learning curve, however, they can then efficiently incorporate systems into their curriculum [6].

An additional challenge that arises with the innovations of content builders lies in the complexity of problem creation. With the focus of the progression of content builders lying in decreasing building time, the ability to design more complex problems with additional capabilities is sometimes lost in the process [7]. The question then arises as to how to strike a balance between providing creators with enough freedom to develop the problems they want, while also optimizing builders for time efficiency and with an approachable learning curve.

2.2. Adaptive Content at Scale

Differentiated instruction, or an individualized prescription of practice is an essential component of modern tutoring systems, producing higher learning gains in experiments in comparison to non-adaptive variants [9]. A main advantage lies in individualizing the tutoring experience, as this type of content can provide students with more appropriate problems conditioned on their skill mastery [10]. Many ITS, such as Cognitive Tutor (now MATHia), have been widely deployed in classrooms where they continue to show significant learning gains over business-as-usual instruction [12]. Tutoring systems, like ASSISTments, are also common grounds for launching content-based learning experiments [11]. These systems could be seen as a type of interactive, adaptive textbook, whereby the student only interacts with content that has not yet been mastered.

Khan Academy hosts one of the largest collections of content for various subjects, distributing video content at scale [13], and could be considered one large, grade-spanning, course. With increased usage due to the recent rise of online learning, Khan Academy as a platform demonstrates the importance of tutoring content existing at such a large scale and that being able to support learners at their own individual pace can be critical for their popularity [14]. Khan Academy secured large amounts of support during its early years, including a two-million-dollar grant [13] and still resorted to AI language learning models to draft some of its content to meet supply demands [31]. Unlike Khan Academy, ALEKS is an ITS, and provides adaptive content through its online service [15]. Grounded in Knowledge Space Theory, ALEKS was developed to provide students with an online tutor focused on personalized content. With proven learning gains comparable to those of classroom instruction, ALEKS distributes hundreds of lessons and content for the K-12 environment [16]. However, in order to sustain such a repository and a service, ALEKS is presented as a premium service [15]. There is a required fee to use the entire collection of ALEKS's content and services, providing yet another instance with a visible financial barrier preventing affordable and simply replicability for a successfully scaled content environment.

Teachers have been successfully crowdsourced to produce content within tutoring systems. The aforementioned system of ASSISTments allows content created for its system to be accessible to anyone through its community features [6]. ASSISTments then uses resources such as student comments to help maintain content, and make minor mistakes, such as spelling errors, available to the content authors [3]. Additional challenges that arise from such a strategy lie in both overcrowding the system with content that is not widely used, as well as lacking a preventative measure to stop incomplete or incorrect content from making it onto the platform in the first place. While revising content after it has been published, and providing a platform for users to give feedback to content creators is important, incorporating content into a curriculum can be difficult for an instructor if there are large amounts of unedited and unsupervised problems, an issue that is reflected amongst many community-focused digital platforms such as Open Educational Resources [17].

2.3. Crowdsourcing and Learnersourcing

The most basic definition of "crowdsourcing" revolves around a "proposer" reaching out to groups of individuals for participation in a "voluntary undertaking of a task", usually in an online environment [18]. Crowdsourcing involves a direct benefit for said individuals, usually in the form of financial compensation, in order to incentivize involvement. Crowdsourcing itself features a set of challenges, as providing an appealing incentive over a large duration of time can be difficult when paired with the effort required to identify and recruit potential participants [19]. However, once these challenges are overcome, it is possible to create a beneficial environment, utilizing the skillset of said participants. To achieve this desired outcome at scale, potential crowdsourcers need to ensure their provided incentive is motivating enough for participants to maintain a long-term involvement within the project, while also making sure to allocate resources and participants in a "flexible manner" [19].

In regard to providing content to educators, Open Educational Resources (OERs) are a key exemplifier of crowdsourcing in education. OERs serve as collections of different types of educational information (from worksheets, to lesson plans, to videos), under either an open license, or in public domain [20]. The crowdsourcing model varies from resource to resource, but there is a united idea of individuals being able to contribute and use content freely. Some OERs feature crowdsourcing incentives for financial sustainability of the platform (referred to as a Donations Model) while others rely on external sponsors to support the openness of the content (referred to as a Sponsorship Model) [21]. While there are additional financial models for different OERs, a main challenge remains that some form of funding needs to exist to keep the resource and the ability to re-distribute content functioning, even if the content itself is free, and the users don't directly contribute to the financial model. Additionally, OERs face the challenge of maintaining high-quality content [17]. With anyone being able to submit their creations in most OER services, high levels of moderation and tools allowing the communities to provide feedback and concerns are necessary to navigate these types of crowdsourced content ecosystems.

With the challenge of a financial barrier persisting in the crowdsourcing ecosystem, as well as the necessity for content to be created by qualified individuals, a subclass of crowdsourcing known as learnersourcing arises. Within learnersourcing, the potential participant is now someone who is particularly knowledgeable of the field, usually taking the form of a recent (or current) learner [22]. The incentive transforms from something usually financial into a form of learning, or reinforcement of a subject, for the participant learner. With the learnersourced task revolving around content the learner is familiar with, involvement in the task allows the display of expertise and creativity on the participants end. This stands to benefit the participant learner through the "Generation Effect", the fact that students recall information better if they have generated it themselves [23]. Thus, learnersourcing stands as a relationship that helps provide expert content without any financial difficulties, while also directly helping learners master a subject.

Learnersourcing serves as a relatively new idea in the crowdsourcing community, so few systems have had the chance to incorporate it. The system PeerWise allows student creation and sharing of "formative practice questions" directly through the system, allowing for instructors to provide student drive supplement to the course material [24]. RiPPLE serves as an adaptive learning platform that incorporates an aspect of learnersourcing, partnering with students to create resources for the system. Within RiPPLE, students can create various questions either by themselves, or in a group, with the ability to also review and customize content from other students [24]. With learnersourcing on the rise, and systems such as RiPPLE being able to incorporate it partially but effectively, the framework appears to be able to create a beneficial ecosystem between learners and content availability at scale.

2.4. Intelligent Textbooks

Outside of classic tutoring systems, attempts to scale content can also be found within intelligent textbooks. Through the assistance of AI as well as frameworks such as Knowledge Space Theory, there has been a push for the creation of adaptive digital textbooks [27]. Textbooks already offer large amounts of accessible and scaled content, so an alternative approach to scaling an adaptive content-base can be to facilitate adaptivity within an already scaled content medium. Utilizing approaches developed for ITS, web-based textbooks have explored the usage of adaptive content presentation and content recommendation [28]. However, many difficulties still remain as to mapping between textbook content and "complex" activities [32].

Machine learning models, such as FACE, have been developed to advance keyphrase extraction and support student modeling, with end goals of creating learning platforms that combine adaptive textbooks with interactive content [28]. This type of student modeling allows implementations of and connections to external content, allowing for the merging of supplementary material onto an intelligent textbook [29]. Paired with early attempts to generate questions based on textbook learning objectives [30], we argue that intelligent textbooks position themselves as a comparable tool to adaptive tutors. Beyond logging, intelligent textbooks facilitate interactivity, instead of a static story about science inquiry, thus allowing for various forms of student personalization. With additional room for research on incorporation of learning-objective focused content into intelligent textbooks [30], they serve as a rigid foundation for scaled content.

3. The OATutor Content Ecosystem

3.1. Introduction to OATutor

Development on OATutor began in late 2019, with the system's open sourcing taking place in April of 2023. OATutor was built upon the foundation of the eight ITS principles, being supported by years' worth of literature showcasing the display and effectiveness of ITS-like systems [12]. OATutor features knowledge tracing-based mastery learning assessment, LTI support for educational systems such as Canvas, and A/B testing capabilities to empower rapid experimentation in the learning sciences.

OATutor was pilot tested in college algebra classrooms in a community college for a total of nine course offerings over the period of five semesters, in which feedback from the course instructor was used to iterate and improve upon the system, resulting in additions to the system's feature list. The open-source release of the tutor includes comprehensive adaptive content for three different levels of algebra, with lessons curated from OpenStax's algebra textbooks [1]. Furthermore, OATutor has displayed noticeable learning gains in early research on the system, when comparing hints generated by the OATutor team to those degenerated by ChatGPT [25].

3.2. Building Content in OATutor

Content creation and curation for OATutor does not require the use of a "builder interface" as present in previously discussed systems. To address the difficulties of instructors having to adapt to new interfaces to utilize tutors in their classroom, as well as to support efficient research capabilities with content, OATutor uses spreadsheets as its basis for building content (Figure 1). Either through Google

Spreadsheets or through Microsoft Excel, individuals create their content on a likely-familiar interface. Even if an individual is unfamiliar with the spreadsheet interface, there are large collections of resources widely available as to how to use them, with the OATutor team also offering OATutor-specific user resources.

<table>
<tr><th>Problem Name</th><th>Row Type</th><th>Title</th><th>Body Text</th><th>Answer</th><th>answerType</th><th>HintID</th><th>Dependency</th><th>mcChoices</th><th>Images</th><th>Variabilization</th></tr>
<tr><td>real11</td><td>problem</td><td>Using a Formula</td><td>A right circular cylinder with radius r and height h has the surface area S (in square units) given by the formula S=2pir(r+h).</td><td></td><td></td><td></td><td></td><td></td><td>https://openstax.org/resources/ba716f0ba5657f6dca83dc7f31ef9af7f93a25ea</td><td></td></tr>
<tr><td>real11</td><td>step</td><td>Find the surface area of a cylinder with radius @{r} in. and height @{h} in. Leave the answer in terms of pi.</td><td></td><td>2*@{r}*(@{r} + @{h}) pi</td><td>algebra</td><td></td><td></td><td></td><td></td><td>r:1|2|3|4|5
h:6|7|8|9|10</td></tr>
<tr><td>real11</td><td>hint</td><td>Substitute</td><td>Substitute @{r}and @{h} into the equation to obtain 2pi(@{r})((@{r})+(@{h}))</td><td></td><td></td><td>h1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>real11</td><td>hint</td><td>Parentheses</td><td>Simplify the parentheses.</td><td></td><td></td><td>h2</td><td>h1</td><td></td><td></td><td></td></tr>
<tr><td>real11</td><td>scaffold</td><td>Parentheses</td><td>What is @{r} + @{h}?</td><td>@{r} + @{h}</td><td>algebra</td><td>h3</td><td>h2</td><td></td><td></td><td></td></tr>
<tr><td>real11</td><td>hint</td><td>Multiplication</td><td>The next step is to simplify multiplication and division.</td><td></td><td></td><td>h4</td><td>h3</td><td></td><td></td><td></td></tr>
<tr><td>real11</td><td>scaffold</td><td>Multiplication</td><td>What is 2*@{r}*(@{r} + @{h})?</td><td>2*@{r}*(@{r} + @{h})</td><td>algebra</td><td>h5</td><td>h4</td><td></td><td></td><td></td></tr>
<tr><td>real11</td><td>hint</td><td>Multiply by pi</td><td>Multiply by pi to obtain 2@{r}@{r}pi + 2@{r}@{h}pi.</td><td></td><td></td><td>h6</td><td>h5</td><td></td><td></td><td></td></tr>
</table>

Figure 1: An example OATutor problem within google spreadsheets

When examining the spreadsheet format, the different sections of a content item are split into columns and rows. The goal of formatting the spreadsheet in this manner was to facilitate a content creation environment designed for creating problems with help features, while also supporting content curation from OERs and textbooks. Thus, the structure found within the spreadsheet attempts to emulate the basic component of a textbook question, while providing additional fields to support the help interface of OATutor. The "Problem Name" column denotes an annotated version of the section title, in an effort to group relevant problems together. The "Row Type" column defines the information each row is providing, separated into "problem, step, hint, scaffold." The spreadsheet interface allows for each hint to have its own dedicated row, allowing for effective implementation of help systems.

The "Title" and "Body Text" columns can be used for any row type. For problems and steps, they reflect the title of the problem, and an explanation, while for hints and scaffolds they dictate the text that appears in the hint interface. The "Answer" and "answerType" columns are used specifically for row types that require user input (steps and scaffolds). "Answer" serves as the correct answer to a given question, and "answerType" indicates how the system treats the answer: "string" answers have to be written exactly by the student, "algebra" answers allow any simplification, and "mc" answers allow for multiple choice. "HintID" and "Dependency" define the ordering of different hints and scaffolds, as well as what prior hints are required to view subsequent ones. "mcChoices" is exclusively used for multiple choice problems and hosts the set of possible solutions. One of the present choices must match the "answer" entry in order to properly work (an automated script checks for such issues during the editing process).

Images can be integrated into the problem through the "Images" column by using a direct image url. Allowing for individual hints and scaffolds to be treated as rows further supports this integration, as the placement of an image can be clearly defined within a problem based on the row it is placed in. The "Variabilization" column allows for the possible inputs of different variables, and variabilization can be applied to everything from body text to mcChoices. The "parent" column (not pictured) is used exclusively for sub-hints and sub-scaffolds and indicates to which previous scaffold such rows belong. Finally, there are three sourcing related columns (also not pictured). "OER src" indicates the respective source of a problem for proper attribution. "KC" refers to the knowledge component as defined in the original source, and "taxonomy" refers to the type of taxonomy the original source utilizes (common core standards, custom format, etc.).

An automated script converts the problems created on the spreadsheet into problems in the system's frontend. At the same time, the script automatically checks if the problem is properly formatted, and provides feedback automatically in the process, emphasizing potential errors in the problem structure. In order to support additional research use cases, the script can be run at a local level as well.

While the spreadsheet interface is simplistic to be approachable, it does not face the aforementioned limitations of design restrictions. With OATutor being opensource, anyone is able to build upon the capabilities of the interface and create structure for any problem type that may be required for their individual research or classroom needs. Thus, it is possible to maintain a user-friendly basic interface, while allowing for complex design if necessitated in specific usage cases of the tutor.

3.3. Content Community

Inspired by the benefits of learnersourcing, the OATutor content team is crowdsourced with a method that resembles the foundations of learnersourcing. Content team members are not recent learners but instead experienced learners with tutoring experience on their respective subject. Because of this, "learning" cannot be the provided incentive for their participation. Instead, content team members are recruited through research apprenticeship programs (such as UC Berkeley's URAP) with the motivation to work in a UC Berkeley research environment, as well as contribute to the open-source nature of the project [1]. As a benefit to the tutor, however, this additional requirement also helps eliminate the risk of low-quality content, as the content building community is comprised of a demographic deeply familiar with the material.

To maintain the content community pipeline, the OATutor content team holds bi-weekly meetings to facilitate discussion on potential difficulties that may arise during content creation and curation. Individual content creators contribute their thoughts and questions to the meeting, with their feedback being used to iterate upon the system and interface design further. These meetings further allow to ensure the fostering of motivation and connection within the team, helping to facilitate a successful crowdsourced environment.

The recruited individuals are split into content creators and editors. Editors are more experienced creators who have been with the team for longer than a semester and are deeply familiar with the content structure of OATutor. Editors review content created by the rest of the team, and make any necessary changes before the content is pushed to the frontend of the system. Furthermore, they provide feedback to content creators, helping facilitate an efficient content pipeline with a rigorous editing and review process. This further protects the quality of the content, while also helping new creators adjust to the system without having to master content creation alone.

Originally, content creators worked on individual spreadsheets, which were then curated into a single editor spreadsheet for review. After editor review, the content was then pushed to a "master document" spreadsheet which was converted to the material on the system's frontend. However, this pipeline was shifted to have content creators work directly on the editor sheet. With an updated automatic script, editors and content creators alike were able to preview what their problems would appear like on the system, allowing for swifter handling of bugs and formatting issues, as well as more immediate feedback on the side of editors.

3.4. Approach to Content Curation

For the creation of high-quality content within the system, and ensuring the content is usable within classrooms, a study of various OERs was conducted to determine an appropriate resource to curate content problems from (with the content team creating hints and scaffolding for said problems). After examining over 25 potential resources for content curation (Figure 2), five main potential candidates were selected (colored in green), with three additional backup resources (colored in yellow). Upon further inspection of the top five resources, OpenStax was selected as the main resource to curate content from.

OpenStax textbooks were used as the foundation for the first three collections of algebra lessons in OATutor. Each collection is classified as a "book" and includes problems for every lesson from the respective OpenStax textbook. Each lesson contains 15-30 problems depending on OpenStax availability, each with custom-generated hints and scaffolds from the content team. Individual problems are tagged with appropriate knowledge components reliant on categorization found within the OpenStax textbooks. This allows for a direct mapping between every individual OATutor problem, and an OpenStax question. Such mapping can also occur at the lesson level, as problems are separated into OATutor lessons based on the respective OpenStax chapter.

OER Name	Website Link	Has Comercial License?	Has Non-Comercial License?	Has Passive Resources?	Has Problem Resources?	Has Problem Resources With Help?
OER Commons	https://www.oer	N/A	Yes	Yes	Yes	Yes
Curriki	https://library.cu	N/A	Yes	Yes	Yes	Yes
National Science Digital Library	https://nsdl.oerc	N/A	Yes	Yes	Yes	Yes
MIT Open Courseware	https://ocw.mit.	N/A	Yes	Yes	Yes	Yes
Khan Academy	https://www.kha	N/A	Yes	Yes	Yes	Yes
BiteScis	https://bitescis.	N/A	N/A	Yes	Yes	Yes
teachit Maths	https://www.tea	N/A	N/A	Yes	Yes	Yes
openstax	https://openstax	Yes	Yes	Yes	Yes	Yes
teachit Science	https://www.tea	N/A	N/A	Yes	Yes	Yes
S.O.S. Mathematics	http://www.sosr	N/A	N/A	Yes	Yes	No
OpenLearn	https://www.ope	N/A	Yes	Yes	Yes	No
BC Capus OpenEd	https://open.bcc	N/A	Yes	Yes	Yes	No
Illustrative Mathematics	http://tasks.illus	N/A	Yes	Yes	Yes	No
ACT Academy	https://actacade	N/A	N/A	Yes	Yes	Yes
CUNY Academy Works	https://academi	Yes, No	Yes, No	Yes	Yes	Yes
Academic Earth	https://academi	N/A	N/A	Yes	No	No
Stem Resource Finder	https://learn.cor	N/A	N/A	Yes	Yes	Yes
Math Resources for Students	https://www.inte	No	No	Yes	Yes	Yes
LUMEN Open NYS	https://courses.	Yes, No	Yes, No	Yes	Yes	No
OASIS	https://oasis.ge	N/A	N/A	Yes	Yes	Yes
iBiology	https://www.ibic	N/A	Yes	Yes	No	No
MERLOT	http://www.merl	N/A	N/A	Yes	Yes	Yes
Math Video Library	https://www.jes	No	No	Yes	Yes	No
QUBES	https://qubeshu	N/A	Yes	Yes	Yes	No
Xpert	https://www.not	No	No	Yes	Yes	Yes
ELA Free Resources	https://www.ela	No	No	Yes	Yes	No
Wisc-Online	https://www.wis	N/A	Yes	Yes	Yes	No
Transum Mathematics	https://www.trar	N/A	N/A	Yes	Yes	Yes

Figure 2: A table showcasing the collection of OERs assessed and the criteria that were reviewed

This one-to-one mapping with the OpenStax curriculum has found great utilization in pilot tests of the system, where OATutor lessons served as supplemental material in a community college classroom to accompany the respective OpenStax textbook material of the curriculum. Additionally, due to the precise tagging of the content and the ability for instructors to assign lessons independently, the content can be incorporated into any respective algebra curriculum, even if said curriculum does not replicate the OpenStax chapter order. With the open-source nature of the content, further potential exists for the content itself to be modified appropriately for any curriculum needs, without the requirement of building new content from the ground up.

4. Applications in Intelligent Textbooks

4.1. Textbook Integration

As discussed regarding its usage with OpenStax textbooks, there is a visible benefit in using an adaptive lesson to supplement a digital textbook. With recent efforts to generate content from digital textbooks showing successful results [26], OATutor can create a bridge between digital textbooks and adaptive content generated from them. The content ecosystem presented above replicated the learning-objective focused content that recent efforts have attempted to create for intelligent textbooks [30], but instead of utilizing machine learning or AI algorithms, the content creation was facilitated by a crowdsourced team of learners.

With a present need for interactive and adaptive content in the field of intelligent textbooks, the OATutor content ecosystem accomplishes curation of three textbooks worth of algebra content with unique hints and scaffolds for the accompaniment of each problem. This demonstrates feasibility of learning-objective focused content generation directly from digital textbooks and creates the opportunity to replicate the content ecosystem for any other necessary textbooks. The created content can be easily integrated and incorporated within textbooks, or classroom curricula centered around them, as exemplified in OATutor's pilot tests [1]. Thus, we argue that creating such an ecosystem and a pipeline for content generation from textbooks is within the spirit of intelligent textbooks.

Within OpenStax specifically, a future incorporation of OATutor can be imagined. It would be reasonable to embed OATutor lessons within the individual OpenStax textbook pages. As OpenStax chapters are structured with theory followed up by problems, and said problems are what has been curated into OATutor, it would be reasonable to feature a section prompting learners to practice the content they have just been exposed to directly within the textbook page. This application and incorporation of OATutor would require minimal work, as the lessons have already been curated.

An alternative method of OATutor embedding could be facilitated by having individual problems within the textbook directly linked to the corresponding tutor questions (or the tutor question completely replacing them). OATutor would track learner mastery throughout the lesson, and would gray out, or disable, subsequent questions of skills that have already been mastered. To avoid concerns about altering the nature of a textbook, for example if the ability to print a physical copy is still desired, OATutor's interactive elements could be enabled in a subtle way that does not change the look of the printed textbook.

Beyond OpenStax, the discussed approaches can be generalized to other digital textbooks. The OATutor approach to crowdsourced content production by transcribing OER content and augmenting it with scaffolding and hints proved sustainable over three years. Such a framework can be applied to other textbooks, and with the open sourcing of OATutor's mastery algorithms, allow for regular digital textbooks to provide adaptive content.

4.2. Other Implementations

Additional pilots of the system could also provide beneficial information about the usability of such content. While supplemental content involves students interacting with the system's lessons, direct integration into curriculums in the form of homework or assessments can provide additional insights into advantages and disadvantages of the current content structure. Facilitation of such usage could also encourage modification of the system's content to more directly reflect assessment requirements of a classroom, allowing for further information collection and research opportunities to examine these necessities and connections to intelligent textbooks.

In terms of future improvements, the curation from OpenStax (and any other future digital textbook) could be iterated upon to automatize the problem curation and require direct intervention from the content team for hint and scaffold creation (serving as the current original work on the team). Furthermore, if the current trends of ChatGPT experimentation for hint generation continue, and the learning gains of ChatGPT help approach those of manual hints and scaffolds, the entire content creation process could be greatly accelerated, and easily integrated with digital textbooks, requiring only editor revision. Such work would require a reformation of the described content ecosystem, and additional studies to equate the learning gains.

5. Discussion and Conclusions

With the entirety of OATutor open-sourced, in addition to a crowdsourced community of learners, the novelty of the content ecosystem should be replicable. Further research is required to determine the time period necessary to fully replicate such an ecosystem, in addition to examining possible modifications and expansions. This includes other organizations and laboratories attempting to replicate the learner-focused crowdsourcing ecosystem of OATutor for their own content creation in addition to creating content at OATutor's scale and efficiency.

Replicability and scaling of the content ecosystem has room to be explored. OATutor successfully created its content community due to systems like UC Berkeley's Undergraduate Research Apprenticeship Program (URAP). Although national programs like the National Science Foundation's Research Experiences for Undergraduates (REU) are still available, relying on such a program to support long-term content ecosystems can be difficult. Even with long term facilitation, there are additional crowdsourcing challenges that still persist, such as cohesiveness. Specifically, the individual voice of each learner is reflected in the problem they help create. Each learner will structure hints and scaffolds in a slightly unique manner, sometimes resulting in a mesh of different or even conflicting methodologies within the same lesson. While OATutor suppressed this issue by assigning one lesson per individual creator, this solution only works at such a granular level, and cannot be replicated for an entire book.

The open sourcing of an adaptive tutoring system with a crowdsourced community of learners and an open content pipeline opens pathways for replicable and novel research to accelerate in the learning sciences. With low-entry-level content building capabilities, but the potential to build upon the system to design content of any complexity, OATutor challenges the current standard for content creation in research environments, while also supporting effective classroom integration through its features. Furthermore, as digital textbooks advance in ways that foster inclusion of active content, OATutor's structure enables opportunities to study and design curricula utilizing the combined capabilities of textbooks and tutors and creates new potential for research on digital textbook integration.

6. Acknowledgements

This paper was peer-reviewed by Zachary Pardos, UC Berkeley, USA. This Word template was created by Aleksandr Ometov, TAU, Finland.

7. References

[1] Pardos, Z.A., Tang, M., Anastasopoulos, I., Sheel, S.K., and Zhang, E. (2023). OATutor: An Open-source Adaptive Tutoring System and Curated Content Library for Learning Sciences Research. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 416, 1–17. https://doi.org/10.1145/3544548.

[2] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. The journal of the learning sciences 4, 2 (1995), 167–207J.

[3] Razzaq, L., Patvarczki, J., Almeida, S. F., Vartak, M., Feng, M., Heffernan, N. T., & Koedinger, K. R. (2009). The Assistment Builder: Supporting the life cycle of tutoring system content creation. IEEE Transactions on Learning Technologies, 2(2), 157-166.

[4] Vincent Aleven, Bruce M McLaren, Jonathan Sewall, and Kenneth R Koedinger. 2006. The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains. In Proceedings of the 8th International Conference on Intelligent Tutoring Systems. Springer, 61–70.

[5] Daniel Weitekamp, Erik Harpstead, and Ken R. Koedinger. 2020. An Interaction Design for Machine Teaching to Develop AI Tutors. ACM, 1–11. https://doi.org/ 10.1145/3313831.3376226

[6] Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education, 24, 470-497.

[7] Leena Razzaq, Jozsef Patvarczki, Shane F. Almeida, Manasi Vartak, Mingyu Feng, Neil T. Heffernan, and Kenneth R. Koedinger. 2009. The ASSISTment Builder: Supporting the Life Cycle of Tutoring System Content Creation. IEEE Transactions.

[8] Weitekamp, D., Harpstead, E., & Koedinger, K. R. (2020, April). An interaction design for machine teaching to develop AI tutors. In Proceedings of the 2020 CHI conference on human factors in computing systems (pp. 1-11).

[9] Huang, G., Qian, X., Wang, T., Patel, F., Sreeram, M., Cao, Y., ... & Quinn, A. J. (2021, May). Adaptutar: An adaptive tutoring system for machine tasks in augmented reality. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-15).

[10] Stern, M. K., & Woolf, B. P. (2000). Adaptive content in an online lecture system. In Adaptive Hypermedia and Adaptive Web-Based Systems: International Conference, AH 2000 Trento, Italy, August 28–30, 2000 Proceedings 1 (pp. 227-238). Springer Berlin Heidelberg.

[11] Ostrow, K. S., & Heffernan, N. T. (2016, April). Studying learning at scale with the ASSISTments TestBed. In Proceedings of the third (2016) ACM conference on learning@ scale (pp. 333-334).

[12] Pane, J. F., Griffin, B. A., McCaffrey, D. F., & Karam, R. (2014). Effectiveness of cognitive tutor algebra I at scale. Educational Evaluation and Policy Analysis, 36(2), 127-144.

[13] Dijksman, J. A., & Khan, S. (2011). Khan Academy: the world's free virtual school. Bulletin of the American Physical Society, 56Dijksman, J. A., & Khan, S. (2011). Khan Academy: the world's free virtual school. Bulletin of the American Physical Society, 56.

[14] Smith, S. J., & Harvey, E. E. (2014). K-12 online lesson alignment to the principles of Universal Design for Learning: the Khan Academy. Open Learning: The Journal of Open, Distance and E-Learning, 29(3), 222-242.

[15] Canfield, W. (2001). ALEKS: A Web-based intelligent tutoring system. Mathematics and Computer Education, 35(2), 152.
[16] Fang, Y., Ren, Z., Hu, X., & Graesser, A. C. (2019). A meta-analysis of the effectiveness of ALEKS on learning. Educational Psychology, 39(10), 1278-1292.
[17] Porcello, D., & Hsi, S. (2013). Crowdsourcing and curating online education resources. Science, 341(6143), 240-241.
[18] Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. Journal of Information science, 38(2), 189-200.
[19] Timothy Shin Heng Mak and Albert Lam. 2022. Two-Stage Auction Mechanism for Long-Term Participation in Crowdsourcing. arXiv preprint arXiv:2202.10064 (2022).
[20] Wiley, D., Bliss, T. J., & McEwen, M. (2014). Open educational resources: A review of the literature. Handbook of research on educational communications and technology, 781-789.
[21] Yuan, L., MacNeill, S., & Kraan, W. G. (2008). Open Educational Resources-Opportunities and challenges for higher education.
[22] Juho Kim. 2015. Learnersourcing: improving learning with collective learner activity. Ph.D. Dissertation. Massachusetts Institute of Technology.
[23] Singh, A., Brooks, C., & Doroudi, S. (2022, June). Learnersourcing in Theory and Practice: Synthesizing the Literature and Charting the Future. In Proceedings of the Ninth ACM Conference on Learning@ Scale (pp. 234-245).
[24] Moore, S., Stamper, J., Brooks, C., Denny, P., & Khosravi, H. (2022, June). Learnersourcing: Student-generated Content@ Scale. In Proceedings of the Ninth ACM Conference on Learning@ Scale (pp. 259-262).
[25] Pardos, Z. A., & Bhandari, S. (2023). Learning gain differences between ChatGPT and human tutor generated algebra hints. arXiv preprint arXiv:2302.06871.
[26] Van Campenhout, R., Dittel, J. S., Jerome, B., & Johnson, B. G. (2021). Transforming Textbooks into Learning by Doing Environments: An Evaluation of Textbook-Based Automatic Question Generation. In iTextbooks@ AIED (pp. 60-73).
[27] Boulanger, D., & Kumar, V. (2019). An Overview of Recent Developments in Intelligent e-Textbooks and Reading Analytics. iTextbooks@ AIED, 44-56.
[28] Chau, H., Labutov, I., Thaker, K., He, D., & Brusilovsky, P. (2021). Automatic concept extraction for domain and student modeling in adaptive textbooks. International Journal of Artificial Intelligence in Education, 31, 820-846.
[29] Rahdari, B., Brusilovsky, P., Thaker, K., & Barria-Pineda, J. (2020). Using Knowledge Graph for Explainable Recommendation of External Content in Electronic Textbooks. In iTextbooks@ AIED (pp. 50-61).
[30] Shimmei, M., & Matsuda, N. (2022). Automatic Question Generation for Evidence-based Online Courseware Engineering. In iTextbooks@ AIED (pp. 18-25).
[31] How does the Khan Academy content team use large language models?. Khan Academy. (2022). https://support.khanacademy.org/hc/en-us/articles/14004618952717-How-does-the-Khan-Academy-content-team-use-large-language-models-
[32] Alpizar-Chacon, I., Barria-Pineda, J., Akhuseyinoglu, K., Sosnovsky, S., & Brusilovsky, P. (2021). Integrating textbooks with smart interactive content for learning programming. In CEUR Workshop Proceedings (Vol. 2895, pp. 4-18). CEUR WS.