

BBookX: Design of an Automated Web-based Recommender System for the Creation of Open Learning Content

Bart Pursel^{‡*}, Chen Liang[‡], Shuting Wang[†], Zhaohui Wu[†], Kyle Williams[‡], Benjamin Brautigam^{*}, Sherwyn Saul^{*}, Hannah Williams^{*}, Kyle Bowen^{*}, C. Lee Giles^{‡†}

[‡]Information Sciences and Technology

[†]Computer Science and Engineering

^{*}Teaching and Learning with Technology

The Pennsylvania State University, University Park, PA 16802, USA

{bkp10,cul226,sxw327,zzw109,kwilliams,bjb40,sps20,hrw115,kbowen}@psu.edu,
giles@ist.psu.edu

ABSTRACT

We describe BBookX, a web-based tool that uses a human-computing approach to facilitate the creation of open source textbooks. The goal of BBookX is to create a system that can search various Open Educational Resource (OER) repositories such as Wikipedia, based on a set of user-generated criteria, and return various resources that can be combined, remixed, and re-used to support specific learning goals. As BBookX is a work-in-progress, we are in the midst of a design-based research study, where user testing guided multiple rounds of iteration in the design of the user interface (UI) as well as the query engine. From an interface perspective, the challenges we present are the matching of the UI to users' mental models from similar systems, as well as educating users how to best work with the algorithms in an iterative manner to find and refine content for inclusion into open textbooks.

Keywords

Information retrieval and extraction, Information seeking and search, personalization, education/learning, Wikipedia

1. INTRODUCTION

Higher Education in the United States and other places is facing an incredible challenge around accessibility and affordability. While a large-scale reduction in tuition seems unlikely in the near future, one area that can see help in the reduction of costs is the use of Open Educational Resources (OERs) [2]. Though OER's adoption is slow to date. One contributing factor is that OER content is found in dozens of repositories on the web. For instructors that want to use OER materials, this creates a challenge of both identifying

quality content from various collections places, then curating it together into a cohesive package for use to support learning.

We describe a novel software platform and system, BBookX [7], which is designed to help instructors searching for OER content, and using a human-assisted computing approach [14] in the identification, curation, and modification of existing OER content. BBookX is a recommender system [12] based on a dedicated information retrieval system that will return specific OER content, based on user-generated search criteria. It is currently using a Wikipedia dump of Aug 5, 2015 which consists of more than 4.9 million English Wiki articles. The user enters into a series of iterations with the recommender system, where he/she accepts and rejects search results. Each time the user loops through the search-accept-reject cycle, the query engine powering the recommender system becomes more accurate. Once the user has identified all relevant OER content, s/he can then begin editing the content, further customizing it for a specific use. More detail about BBookX can be found in [7, 8]. To our knowledge, this is the first automated book building tool.

The vast majority of OER content creation initiatives follow a traditional publishing model; authors are identified to create content, content is sometimes peer-reviewed, then the content is published or made available on the web with a corresponding usage license. BBookX is unique in that it leverages a user interface (UI) and query engine to greatly expedite the creation and sharing of quality open source textbook materials, breaking free of traditional publishing

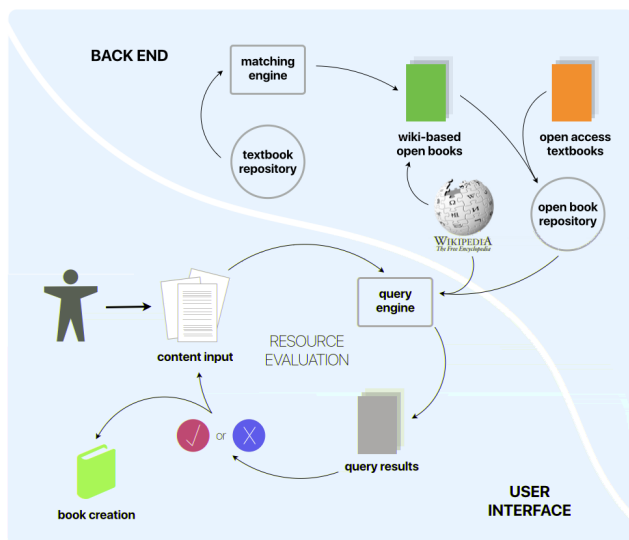


Figure 1: Systems model of BBookX.

2. SYSTEM OVERVIEW

The system overview of BBookX shown in Figure 1 consists of two major components: the backend and the UI. The backend consists of indexed books and OER content using Solr/Lucene [5]. The backend also contains the query engine used to take user-generated search criteria, and search the index for relevant results. The UI is where users create, edit, and store books, and also work with the recommender system. The content that powers the recommender system currently comes from Wikipedia, as it is the largest, open source of OER content available at this time. BBookX is built in such a way that additional OER repositories can be added as they emerge, such as OpenStax² and Saylor Academy³, creating a growing ecosystem of content that can be combined in interesting ways to create new, open books.

The interactive book creation component allows users to specify the information of the book that they want to build using any type of unstructured text. The query engine then uses this text to search and retrieve a list of indexed educational resources ranked by the relevance to the query. An interactive user interface provides easy selection and drag/drop functions allowing users to evaluate the returned resources. User feedback is utilized by an explicit relevance feedback mechanism [13] to reformulate the query to generate a new list of results. The generated book is refined through this interactive search process.

Users have the ability to create many books in BBookX, creating a personal library of open books. Using methods similar to those employed in VizRec [10]. BBookX hopes to also take into account individual user preferences, based on existing books belonging to the user, in the recommendation of future content. Very similar to how Amazon's recommendations increase in accuracy the more items a user purchases from Amazon, BBookX will employ a similar approach, taking into account past books a user created, that will influence present and future searches by the same user.

²OpenStax (<https://openstaxcollege.org/>)

³Saylor Academy (<http://www.saylor.org/books/>)

3. DESIGN-BASED RESEARCH

As BBookX matures, the use of design-based research will guide its development. Design-based research is an established empirical methodology for instructional designers interested in changing instructional outcomes and enhancing the learner experience through development of technologies that support learning in various environments [3]. Design-based research documents key design decisions, often influenced by user testing, as well as leads to new principles adopted by designers, as they iteratively work through designs. To date, twelve user-tests were conducted, that helped us move through two iterations of the BBookX platform, and provide additional data driving the third iteration. Each user was asked to create a 3-chapter book associated with one of the user's courses. A researcher was present during these when the instructors participated in these activities, observing instructors using BBookX, taking notes, and encouraging the instructors to think out loud, verbally articulating how they navigate BBookX, and their expectations when using different sections of the software. The following sections present user testing data and subsequent design iterations.

4. FIRST ROUND OF ITERATION

The original interface was primarily developed for a single webpage. After logging in and clicking "Create Book", users were presented with a page that began by creating a book title and description, and then users added chapters. Early feedback from the design team, as well as a test user, quickly led to the flattening of the interface, breaking down steps of the book creation process to different webpages. For example, book creation was moved to a single page, and each chapter was created and modified on a separate webpage, compared to all the interactions related to a book occurring on one large webpage.

5. SECOND ROUND OF ITERATION

The second iteration of the interface was tested with seven faculty members, across five disciplines. Each faculty member was asked to use BBookX to create a three-chapter book, associated to a course s/he regularly taught. The researcher then observed the faculty member, asking s/he to think aloud while using BBookX, in order for the researcher to gain a better understanding of the thought processes of faculty while using the software. The first challenge observed was the use of the two text fields. When a user clicks "Add Chapter", s/he is presented with two text entry fields: Title and Keywords and Phrases (see Figure 2).

In terms of the query engine, it was originally developed to use both fields to add to the search criteria, placing a heavier weight on the data entered in the "Title" field. Through testing, we discovered our users often had trouble reaching a diversity of search results. For example, if a user created a title 'Artificial Intelligence', and added keywords to the following text field such as 'machine learning', 'intelligent agent', and 'natural language processing', the query engine would return results almost entirely based on the title, 'Artificial Intelligence', and it would take users several cycles through the search process to find results based on keywords entered in "Keywords and Phrases".

This quickly led to a revision of the interaction between the UI and the query engine. Most testers used a very

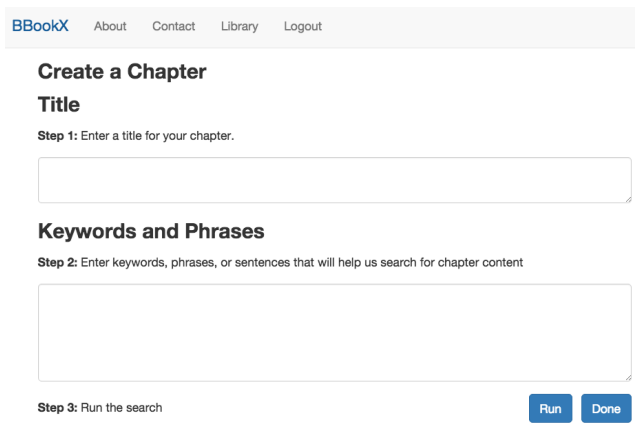


Figure 2: Title and Keywords and Phrases fields.

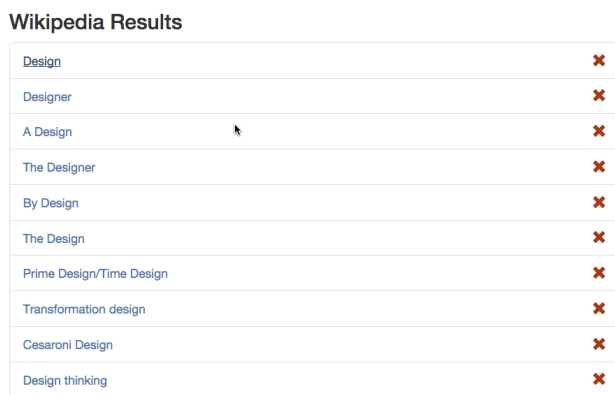


Figure 3: The original display of search results.

generic or umbrella term for a chapter title, then users added much more granular terms and phrases in the second text field. The first change we made was to weight the title and keywords fields the same in the query engine. This still did not lead to a diversity of results users were seeking. The next change was to eliminate entirely the data entered in the “Title” field in terms of adding it to the query engine, and only using the data entered in the “Keywords and Phrases” in the query engine. The query engine then began to return a diversity of results much more pleasing to users.

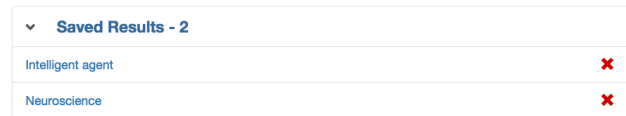
Another challenge was how users worked with search results. Results were presented in sets of ten, with a large, red “X” to the right of each result. The design intent was to allow users to delete or remove an entry by clicking on the “X” beside a result (see Figure 3). When a user ran his or her second query within a chapter, the query engine kept all results that were not removed, and retrieved a set of new results to replace those that were removed by the user. This led to some confusion, as the results that remained from the first search were still embedded in the list of new results that that were returned from the second search within the same chapter. If users performed multiple searches, they often were confused on which results they saved from early searches, and which results were new.

The current UI was changed so that instead of removing results by clicking a red “X”, users now keep results by click-

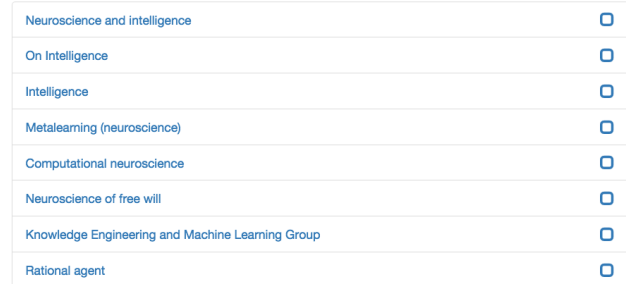
ing a checkbox (see Figure 4). Additionally, the results that are kept are now moved to a new area of the UI, to aid the user in keeping track of what results s/he kept throughout multiple searches within a chapter.

Step 4: Click the links to preview each result in a new browser tab

Step 5: Click the for results you wish to keep



Wikipedia Results



Step 6: After selecting all the results you wish to keep, re-run the search or modify the search terms.

Regenerate Results

Not happy with the results? Modify your search

Figure 4: Improved method for displaying and interacting with search results.

6. FIELD TEST

After the second round of iteration, BBookX was deployed with students in an introductory information sciences and technology course. A lab assignment was created that required the use of BBookX, asking the students to use the software to generate a 3-chapter book, covering topics at the intersection of each students’ major and information sciences. An exploratory survey was then administered (n=140), used to better understand the usability of BBookX, On a four point scale (very positive, positive, negative, very negative), 72% rated their overall experience with BBookX either positive (59%) or very positive (13%), 89% rated the learnability of BBookX either positive (45%) or very positive (44%), 65% rated their satisfaction with BBookX either positive (55%) or very positive (10%), and 62% rated the efficiency of using BBookX either positive (46%) or very positive (16%). During a post-lab discussion in class, it was discovered that some of the students did not realize that the searches within each chapter are designed to be iterative; the more you accept or reject results, the more accurate the successive search. This requires further exploration, as one hypothesis on why some students found the efficiency and satisfaction of using BBookX to be negative is that they only ran a single search within each chapter, thus not experiencing the results getting more personalized based on user actions.

The most interesting question asked students if BBookX surfaced interesting pages of content, including things the student did not know before completing this homework. On a 4-point scale (strongly agree, agree, disagree, strongly disagree), 73% answered either agree (61%) or strongly agree

(12%). This speaks positively about the algorithms powering the software, as they are finding and surfacing pages of content that provide new learning opportunities for students focused on their specific majors. Of note is that this was a general education course, with a wide range of class standings, from freshman to 5-th year seniors.

7. UPCOMING THIRD ROUND OF ITERATION

The current version of BBookX was tested by three faculty members and a graduate student, with more testing scheduled. The query engine is currently returning interesting and relevant results, as judged by testers, although in a very small number of instances a user can not seem to get the query engine to return a page that s/he knows exists in Wikipedia. This may be a problem related to the frequency of indexing that we are currently exploring.

From the UI perspective, users experience the most confusion while working with results. The problem cited by one user is the mental model of search; most users enter BBookX with a mental model associated with Google. Mental models provide a way of understanding a user's motivation and thought-process when engaging in an activity [16]. In this case, the use of Google's search engine is influencing how the user expects BBookX to function. But BBookX differs in that after a user performs a search, if s/he does not elect to keep certain results, when the user runs the search for the second time, all results not kept are eliminated from the potential result candidates in that specific chapter. This causes problems when, for example, a user runs a search, and receives two results s/he finds interesting, and eight unwanted results. One of our users immediately modified the search criteria, and ran the search a second time, without realizing that the two results s/he found interesting were now removed from the results pool. Again, the mental model of search engines such as Google is causing some confusion with users. Possible solutions we plan to test include moving the results closer to a Netflix or Facebook mental model, where instead of just asking a user to check a box to keep a result, the user will be asked to 'like/dislike' or 'thumb up/thumb down' each result. This solution may also help communicate to users that the query engine is designed to be used multiple times per chapter, adapting to the user each search cycle, based on user inputs.

In general, the process of teaching users how to best interact with the query engine is a significant challenge. For example, in order to get accurate results quickly, users should add a great deal of search criteria in the keywords and phrases text field, such as course objectives and presentation notes. Though most users only enter a very small number of search criteria, limiting the number of accurate results per search. The query engine is also built to adapt to user's search actions, such as accepting and rejecting specific results. The more search cycles a user performs within each chapter, the more accurate the query engine becomes. Some of our early users only ran one search per chapter, not realizing that multiple search cycles per chapter will often yield better results.

8. BOOK PUBLISHING

Once a user finalizes his or her book, BBookX features a book generation tool. Currently, book generation involves

IST 110: Introduction to Information Sciences and Technology

<ul style="list-style-type: none"> ▼ Chapter 1: Information Sciences Information science Internet History of the Internet
<ul style="list-style-type: none"> ▼ Chapter 2: Computer Hardware, Software, and Networking
<ul style="list-style-type: none"> ▼ Chapter 3: The Internet of Things
<ul style="list-style-type: none"> ▼ Chapter 4: Web Search

Figure 5: Example of a completed book, when embedded on a website. The first chapter is expanded, to view the links to each section.

extracting text-based content from the selected Wikipedia entries, and combining them in a text file with some formatting. This feature was conceptually well-received by user testers, though it quickly was apparent that distributing a lengthy text file to students as their primary textbook was undesirable. The second phase of development planned for BBookX involves a new publishing UI that users will interact with after finalizing a book. The publishing UI is intended to be a flexible UI that allows users to add, edit, and delete content from the generated book. Once finished in the publishing phase, the book can be shared with students in a variety of methods, such as a designed web interface, an e-text, PDF, and possibly other formats.

The current implementation to quickly make the finalized books more accessible is to use a 'Share' feature, which creates an iframe containing a link to the final book, and users can embed this iframe on websites and in Learning Management Systems (LMS)(see Figure 5). This creates a quick way to share created books, though users do not have any ability to edit books. One user is leveraging a book created in this way as the primary textbook to support a course of 150 students.

9. DESIGN LESSONS

An important aspect of design-based research is to share lessons learned throughout the design process that both shape us as designers as well as informs design communities at large. Throughout the design and development process of BBookX, several design lessons emerged, some of which we are still working through. The first lesson involves a flat vs. vertical design of the UI. The original UI was vertical, such that all user actions within a single book were completed on a single webpage, leading some users to have very lengthy pages, with a great deal of information to manage simultaneously. The move to a more flat interface was a well-received change by users, allowing a user to work within a specific chapter at a time with little cognitive overhead. The vertical UI was much quicker to prototype, though the developer then needed to re-work the UI into a flat model, taking additional development time. Our current UI may even be flattened one more time, moving the search interface and re-

sults interface to two separate webpages. In future projects that require the user to manage various types of data and interactions throughout a system, the design will begin much closer to a flat design, breaking up the system into smaller chunks that are easier to manage for the end user.

From an OER perspective, we anticipated that users would want the ability to modify books, specifically to add content such as images, videos, and transitions between chapters and sections. Users did appreciate this feature, though a much more immediate need was the ability to delete content that was returned within specific search results. For example, some Wikipedia entries are very lengthy, and a user may only want to use a percentage of the overall page. The ability to easily delete content returned from Wikipedia is much more important in the eyes of our users compared to the ability to add their own content to books. From a design perspective, it shifts our focus from creating a UI allowing users to produce and embed more interesting content, to a UI that allows users to quickly identify and remove content.

As this project took shape, the primary focus was on functionality of the UI and the query engine. After basic functionality was in place, we then focused on design. The first two designs of the UI unfortunately did not take into account existing mental models of end users; we simply followed basic design principles. While this was a good start, in the future selecting a prevalent mental model that target end users already have familiarity with should drive the design of the UI from the start of the project.

10. CONCLUSION AND FUTURE WORK

We present an early description and a design narrative of BBookX, a novel recommender system designed to create open textbooks. Compared to existing recommender systems based on techniques such as collaborative filtering [6, 11], the system is unique in that it uses a human-assisted computing approach, leveraging both the strengths of computing, as well as the strengths of humans, to quickly build accurate open textbooks. Future work includes creating concept hierarchies for books, to identify prerequisite content as part of the book creation process [1, 9, 15]. Personalized learning is also a possible direction, similar to the approach described by Garrido and Onaindia [4], where BBookX can take into account prior knowledge, and present book materials in a specific order, based on the learning gaps and trajectories of each student. Please see [8] for a demo and related paper of the working tool.

11. ACKNOWLEDGEMENTS

We gratefully acknowledge partial support from the National Science Foundation.

12. REFERENCES

- [1] E. Cambria and B. White. Jumping nlp curves: a review of natural language processing research. *Computational Intelligence Magazine, IEEE*, 9(2):48–57, 2014.
- [2] T. Caswell, S. Henson, M. Jensen, and D. Wiley. Open content and open educational resources: Enabling universal education. *The International Review of Research in Open and Distributed Learning*, 9(1), 2008.
- [3] T. D.-B. R. Collective. Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, pages 5–8, 2003.
- [4] A. Garrido and E. Onaindia. Assembling learning objects for personalized learning: an ai planning perspective. *Intelligent Systems, IEEE*, 28(2):64–73, 2013.
- [5] T. Grainger and T. Potter. *Solr in action*. Manning Publications Co., 2014.
- [6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.
- [7] C. Liang, S. Wang, Z. Wu, K. Williams, B. Pursel, B. Brautigam, S. Saul, H. Williams, K. Bowen, and C. L. Giles. Bbookx: An automatic book creation framework. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, pages 121–124. ACM, 2015.
- [8] C. Liang, S. Wang, Z. Wu, K. Williams, B. Pursel, B. Brautigam, S. Saul, H. Williams, K. Bowen, and C. L. Giles. Bbookx: Building online open books for personalized learning. In *AAAI*, 2016.
- [9] C. Liang, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1668–1674. ACL, 2015.
- [10] B. Mutlu, E. Veas, C. Trattner, and V. Sabol. Vizrec: A two-stage recommender system for personalized visualizations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, pages 49–52. ACM, 2015.
- [11] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.
- [12] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
- [13] J. J. Rocchio. Relevance feedback in information retrieval. 1971.
- [14] D. Shahaf and E. Amir. Towards a theory of ai completeness. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, pages 150–155, 2007.
- [15] P. P. Talukdar and W. W. Cohen. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 307–315. Association for Computational Linguistics, 2012.
- [16] I. Young. *Mental models: aligning design strategy with human behavior*. Rosenfeld Media, 2008.