BBookX: Building Online Open Books for Personalized Learning

Chen Liang[†], Shuting Wang[‡], Zhaohui Wu[‡], Kyle Williams[†], Bart Pursel[*],
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
{cul226,sxw327,zzw109,kwilliams,bkp10,bjb40,sps20,hrw115,kbowen}@psu.edu, giles@ist.psu.edu

Abstract

We demonstrate BBookX[1], a novel system that automatically builds in collaboration with a user online open books by searching open educational resources (OER). This system explores the use of retrieval technologies to dynamically generate zero-cost materials such as textbooks for personalized learning.

Introduction

Textbooks play a crucial role in both the teaching and learning process. However, in practice, textbooks have several issues. Authoring a new textbook usually requires a great amount of time and effort, even for experts. A tool that facilitates the book creation process could be very helpful. It is not easy to keep an existing textbook up-to-date, especially for fast changing domains such as computer science. For example, most existing machine learning books as of this date lack recent developments in deep learning. However, most of such information can be found in OER, such as publicly available scientific papers, lecture notes, Wikipedia, etc. For certain classes, teachers may want to modify the book content, change the order of sections, or combine content from different books. This can give flexibility for designing personalized textbooks.

To deal with these issues we propose BBookX, a novel collaborative computer facilitated textbook creation system. Designed to utilize information retrieval techniques to intelligently harvest existing OER, BBookX is currently built on top of Wikipedia. However, it can also incorporate other available OER.

BBookX has an interactive Web interface that supports real-time book creation. Given a set of user-generated criteria, it returns a list of relevant material that can be put into the book. Users can accept, reject, or reorder the returned results. Such feedback allows the system to reformulate the query in order to return better results. In addition, BBookX can create an open version of existing textbooks by automatically linking their existing book chapters to Wiki articles.

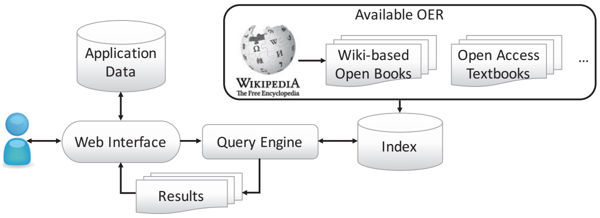


Figure 1: System overview.

BBookX System Overview

Currently, BBookX is a Web-based application, as shown in Figure 1, and interacts with users through a Web interface. BBookX includes the following components:

OER Retrieval

Since BBookX is designed to make use of existing OER, it is important to collect high-quality and up-to-date open resources. The application of Wikipedia, the largest open encyclopedia, in education has long been studied (Seely Brown 2008). It also has wide coverage and high editing frequency. BBookX is currently using a Wiki dump of Aug 5, 2015 which consists of more than 4.9 million English Wiki articles. Besides Wikipedia, we are also collecting through web crawling existing online open access textbooks. Available data sources include Wikibooks[2], Saylor Academy[3], OpenStax College[4], etc. However, since the number of free online textbooks collected is very small compared to the size of Wikipedia, we have not yet included these resources.

We have also explored BBookX's ability to generate with Wikipedia open textbooks that are based on high quality existing textbooks (Wang et al. 2015). We see this as a way to repurpose existing high-quality and well-structured knowledge to make it more readily accessible.

Indexing

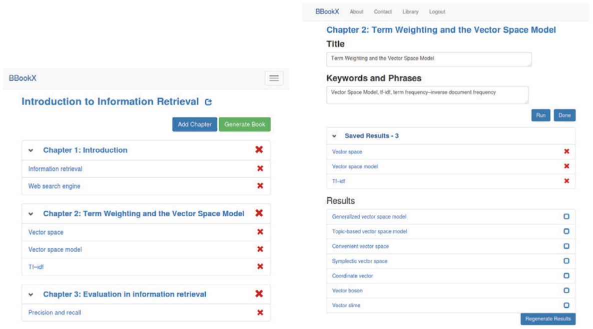
In order to provide efficient retrieval of collected OER, we create a local searchable version. OER are first prepro-

[1] https://book.tlt.psu.edu/

[2] https://en.wikibooks.org

[3] http://www.saylor.org/books/

[4] https://openstaxcollege.org/



Introduction to Information Retrieval

Chapter 1: Introduction
- Information retrieval
- Web search engine

Chapter 2: Term Weighting and the Vector Space Model
- Vector space
- Vector space model
- Tf–idf

Chapter 3: Evaluation in information retrieval
- Precision and recall

(a) Building a book.

Chapter 2: Term Weighting and the Vector Space Model

Title: Term Weighting and the Vector Space Model

Keywords and Phrases: Vector Space Model, tf-idf, term frequency–inverse document frequency

Saved Results - 3
- Vector space
- Vector space model
- Tf–idf

Results
- Generalized vector space model □
- Topic-based vector space model □
- Convenient vector space □
- Symplectic vector space □
- Coordinate vector □
- Vector boson □
- Vector slime □

(b) Adding a chapter.

Figure 2: Web interface.

cessed, which includes tokenization, stop word and punctuation removal, conversion to lower case, and stemming. Apache Solr[5] builds a full text index for the content of each document. In this phase we also extract the keyphrases of each document using the Maui tool (Medelyan, Frank, and Witten 2009) and index them to compute a similarity score.

Querying

Given the user-specified description q about a chapter, BBookX will first query q in the pre-built index and return a set of candidate relevant OER. For each candidate resource d, BBookX calculates the similarity score between q and d based on a weighted combination of title similarity, content similarity, and keyphrase similarity. Then a ranked list of relevant OER is returned to users.

So users can modify the results, BBookX allows them to decide whether to accept or reject any returned resource. A relevance feedback mechanism is incorporated in the system to reformulate the query based on users' judgements about returned results. Specifically, the top 20 keyphrases from resources kept by users are selected based on term frequency and used for query expansion. These keyphrases are added to q to form a new query q'. Users' feedback is especially helpful when the query includes words that have multiple senses, such as "apple", "set", etc. More details of the BBookX's backend can be found in (Liang et al. 2015a).

Web Interface

The software for BBookX is a node application built with the SANE stack (Sailsjs and Emberjs). Sailsjs is used for the API interface to store in MongoDB the application data, including information about users and their built books. Emberjs is used to build the JavaScript front end.

The current web interface components are shown in Figure 2. To use BBookX, users start with registration and login. They can then choose to keep working on previously built books in the "Library" or start building a new book. The book building process first requires adding a book title. As shown in figure 2a, users can then click the "Add Chapter" button to work on a new chapter. Figure 2b shows the interface to add a chapter. After creating a chapter title and adding a short description which can be keyphrases or sentences, users click the "Run" button to let BBookX retrieve a list of relevant OER. When the system finishes searching, users can review each result by clicking the links that allow a preview in a new browser tab. Users then click the results they wish to keep and add them to the "Saved Results" list. If users are still not satisfied with the results or want more relevant OER, they can click "Regenerate Results" to re-run the search. User feedback is be utilized by the system to reformulate the query for other results. In addition, BBookX supports reordering OER or chapters by dragging and dropping. The built books can be exported as text files for further editing or as HTML for embedding in other Web pages.

Existing Systems

To our knowledge, other existing textbook building systems do not support the automatic retrieval and organization of OER, but require manually attached related resources. For example, Wikibooks offers a wiki-based platform allowing users to create books collaboratively. FlexBook[6] is a textbook authoring platform where users produce and customize the book content by re-purposing educational content.

Conclusions and Future Work

We introduced BBookX, a novel collaborative automatic text book building system designed that makes use of available open educational resources and applies simple but useful information retrieval methods that facilitate a textbook authoring process. Currently we are working on incorporating prerequisite relations among concepts (Liang et al. 2015b) into BBookX, which aims to increase the coherence of the book.

Acknowledgments

We gratefully acknowledge partial support from the National Science Foundation.

References

Liang, C.; Wang, S.; Wu, Z.; Williams, K.; Pursel, B.; Brautigam, B.; Saul, S.; Williams, H.; Bowen, K.; and Giles, C. L. 2015a. Bbookx: An automatic book creation framework. In *Proceedings of DocEng*, 121–124. ACM.

Liang, C.; Wu, Z.; Huang, W.; and Giles, C. L. 2015b. Measuring prerequisite relations among concepts. In *Proceedings of EMNLP*, 1668–1674. ACL.

Medelyan, O.; Frank, E.; and Witten, I. H. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of EMNLP*, 1318–1327. ACL.

Seely Brown, J. 2008. Open education, the long tail, and learning 2.0. *Educause review* 43(1):16–20.

Wang, S.; Liang, C.; Wu, Z.; Williams, K.; Pursel, B.; Brautigam, B.; Saul, S.; Williams, H.; Bowen, K.; and Giles, C. L. 2015. Concept hierarchy extraction from textbooks. In *Proceedings of DocEng*, 147–156. ACM.

[5] http://lucene.apache.org/solr/

[6] http://www.ck12.org/