Augmenting Course Material with Open Access Textbooks

Smitha Milli and **Marti A. Hearst**
University of California, Berkeley
Berkeley, CA 94720
smilli@berkeley.edu, hearst@berkeley.edu

Abstract

Online, open access, high-quality textbooks are an exciting new resource for improving the online learning experience. Because textbooks contain carefully crafted material written in a logical order, with terms defined before use and discussed in detail, they can provide foundational material with which to buttress other resources. As a first step towards this goal, we explore the automated augmentation of a popular online learning resource – Khan Academy video modules – with relevant reference chapters from open access textbooks. We show results from standard information retrieval weighting and ranking methods as well as an NLP-inspired approach, achieving F1 scores ranging from 0.63, to 0.83 on science topics. Future work includes taking into account the difficulty level and prerequisites of a textbook to select sections that are both relevant and reflect the concepts that the reader has already encountered.

1 Introduction

A learner who is studying material from an online course, such as a video from a Khan academy physics sequence, may desire additional reading material to supplement the current video or exercise. It can be distracting to do a web search to find relevant material, and furthermore, the material that is found may be described at the wrong level or may assume prerequisite knowledge that the learner does not have. To this point, Mathew et al. (2015) note that online encyclopedic resources, such as Wikipedia, are not pedagogically organized and tend to have many cyclic dependencies among articles.

Textbooks written for students are specifically designed for learning. Material is carefully organized to define terms before use or to point the reader to the location in which the material will be discussed in more detail. Content is described at a consistent reading level and notation and formatting are also consistent.

However, to date, textbooks have not been widely used for automated online recommendations, most likely because they have not been freely available online for many subjects. This situation is changing with the advent of projects like OpenStax (Pitt, 2015)[1] for which respected educators are writing and vetting free online textbooks in major subject categories.

In this work we explore the potential of augmenting online course materials – specifically Khan Academy modules[2] – with relevant supplemental reading from textbooks. We show that even very simple algorithms can go a long way towards making effective recommendations.

2 Related Work

There has been some related work in aligning textbook content to other content. Contractor et al. (2015) identify the need to automatically label instructional materials with learning standards, which are defined hierarchically from general goals down to lists of instructions that define the skills that students should learn during a course or within a curriculum. They develop an algorithm for representing the content within a list of learning standards

---

[1] https://openstaxcollege.org
[2] https://www.khanacademy.org

for high school math and science curricula and label corresponding portions of "educational documents" and Khan academy video transcripts. They use an unsupervised method that models each instruction as a collection of terms that are relevant to that instruction and use external resources including Wikipedia, Wordnet, and a word vector embedding algorithm trained on Wikipedia and news text for term expansion. They allow a match between a text and an instruction only if the higher level goals also match. When associating learning goals with educational documents, they achieve accuracy of 81% for math and 71% for science.

Textbooks often refer to the same concept in multiple locations, and require the reader to make digressions to other parts of the text to understand concepts that they are not familiar with. Agrawal et al. (2013) address the problem of automatically determining which concepts described elsewhere in the textbook are most relevant to what the reader is viewing at the current juncture. They create a model of the structure of references to concepts within sections of the textbook and model the manner in which readers would navigate these references based on their structure within the book. The model does not examine the text itself.

Agrawal et al. (2011), working with substandard textbooks (written in a developing nation), identify the sections can be enriched by better written content. They define a syntactic complexity score that makes use of the maturity of the text and a semantic dispersion score based on the observation that sections that discussed concepts with respect to one another were of higher quality. Their earlier work (Agrawal et al., 2010) linked textbook content to web resources. Our intent starts with the opposite assumption: that the textbooks are authoritative and are to be linked to other content.

Mathew et al. (2015) distinguish between pedagogic and general resources such as thesauri, noting that the latter have good coverage but are not structured to aid in learning. They assess a graph-theoretic algorithm for collapsing word definitions into more compact forms.

3 Methods

Khan Academy *modules* are courses that cover broad subjects such as "Physics", "Chemistry", "Biology" and are broken down into *submodules* focusing on more specific topics within the subject. Each submodule consists of some combination of videos, readings and interactive exercises presented within a dynamic web interface. For example, within the physics module on Khan Academy, submodules include "Force and Newton's laws of motion", "Magnetic forces and fields", etc. See Figure 1 for a screenshot.

Our goal specifically is: Given a Khan Academy module and a textbook, for each submodule in the Khan Academy module, assign the chapters from the textbook that teach the same concepts as the submodule. We wish to label each of these submodules with relevant chapters for reading. We present three methods to do so.

3.1 Method 1: TF-IDF document similarity

We use a standard method for document similarity comparison from information retrieval: weighting terms with tf-idf scores, converting documents into vectors with these weights, and comparing documents by taking the cosine similarity of the vectors (Baeza-Yates et al., 2010). Each submodule is represented using the text from its main page, which only consists of titles and short descriptions of videos, readings, and exercises. The text from the exercises and video transcripts were not used.

The vocabulary of words extracted consists of all words in the submodules excluding stopwords and terms with a document frequency over 0.9. Each submodule and chapter is encoded as a vector of these words using tf-idf weights computed on the set of submodules. Let D be the set of submodules, N_d be the number of words in submodule d, $f_d(t)$ be the number of times term t appears in submodule d. Term frequency is computed as the raw frequency of a term in all submodules, i.e. $\frac{\sum_{d \in D} f_d(t)}{\sum_{d \in D} N_d}$. The inverse document frequency for a term is computed as $\log \frac{|D|}{|\{d \in D : t \in D\}|}$. For each submodule the chapter with the highest cosine similarity is selected.
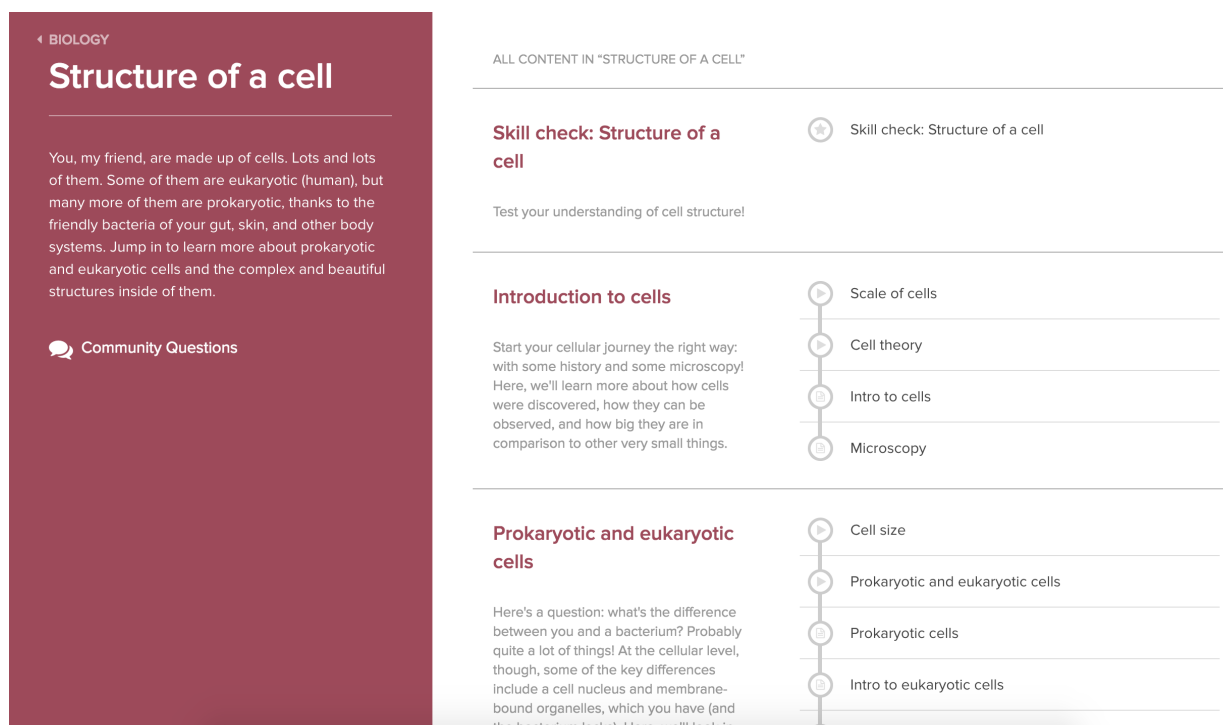
Figure 1: A screenshot of the "Structure of a cell" submodule from the Khan Academy biology module.

3.2   Method 2: Learning objective frequencies

Although computing document similarity works well when we assume a 1-1 correspondence between a submodule and chapter in a textbook, some submodules may span multiple chapters or no chapters at all.

To address this issue we create a method based on learning objectives. A *learning objective* for a submodule is a concept that is taught in the submodule with the goal of being understood by a learner after completion of the submodule. In this work we assume that learning objectives can be represented by key phrases corresponding to new terms that are taught to the learner such as acceleration, cell division, photosynthesis, etc. This is a very simple representation compared to, say, a knowledge-based method.

Our method extracts learning objectives from a Khan Academy submodules and searches for which chapter teaches those learning objectives with the understanding that different learning objectives may be taught in different chapters. Essentially we are reducing the assumption of a 1-1 correspondence between a submodule and chapter to a 1-1 correspondence between a learning objective and a chapter.

For each submodule a list of learning objectives is extracted. The chapters assigned to the submodule consists of the set of chapters assigned to each learning objective. The pseudocode for this general algorithm is the *augmentSubmodule* procedure in Algorithm 1. *augmentSubmodule* depends on two components: the extraction of learning objectives, *extractLearningObjectives*, and the assignment of chapters to learning objectives, *pickChapterForObj*.

In this work *extractLearningObjectives* is implemented as a keyphrase extraction. The keyphrase extraction is a rule-based approach that breaks up lists and terms that are separated by the word 'and'. For example, the submodule of the physics module titled "Electric charge, electric force, and voltage" contains the keyphrases "electric charge", "electric force", and "voltage", which the algorithm extracts.

In addition, the words in the title are tagged with parts of speech, so that the pattern *"JJ$_1$ and JJ$_2$ NN"* extracts both *"JJ$_1$ NN"* and *"JJ$_2$ NN"*. For example, both the phrases 'balanced forces' and 'unbalanced forces' are extracted from 'balanced and unbalanced forces'. Terms can be filtered to those that occur at least a minimum frequency in the textbook.

Our implementation of *pickChapterForObj* is the

procedure *pickChapterForObjFreq* as shown in Algorithm 1. Let the set of chapters be denoted C. $f(t, c_i)$ is the frequency of a term t in chapter c_i. The chapter picked for a learning objective t is the chapter with the highest frequency, $\text{argmax}_i f(t, c_i)$ for $i \in \{1, 2, \ldots, |C|\}$.

3.3 Method 3: Learning objective spikes

Method 3 is based on the notion of learning objective spikes and is the same as Method 2 except with a change in how a chapter is assigned to a learning objective. We say a learning objective has a *spike* in chapter i if it has a sudden increase in probability in chapter i compared to any of the previous chapters. The threshold for what counts as a "sudden increase" can be tuned. In Method 3 the chapter picked for a learning objective is the chapter with the first spike (if any exist) for the learning objective.

The motivation for this spike-based method is that because textbooks are written to teach, in most cases, when a new term is first discussed in detail is where it is defined and explained best. The assumption of the spike method is that this definition chapter in most cases most useful to show to a learner, even though some following chapters may use that term more frequently in the context of describing some more advanced concepts.

For example, as shown in Figure 2, the word "voltage" is defined in chapter 19, where a spike is seen, and then used many times in chapter 21 in a discussion of circuits. However, "voltage" is mentioned only in passing in chapters 17 and 18, and so those are not the best chapters to show the learner as compared to chapter 19, which has a spike in usage. Thus in Figure 2, Method 3 performs better than Method 2.

Let the set of chapters be denoted C. $f(t, c_i)$ is the frequency of a term t in chapter c_i. The probability of chapter c_i given a term t is $p(t, c_i) = \frac{f(t, c_i)}{\sum_{j=1}^{|C|} f(t, c_j)}$. The score for a chapter is $s(t, c_i) = p(t, c_j) - \max_{1 < j < i} p(t, c_j)$. Finally, the chapter assigned to a term is chosen by picking the smallest i such that $s(t, c_i) > P$ is true where P is a tunable threshold. The algorithm fails to identify a chapter for the term if $\forall i \ s(t, c_i) \leq P$. The final algorithm for Method 3 is the *augmentSubmodule* procedure using the *pickChapterForObjSpike* procedure
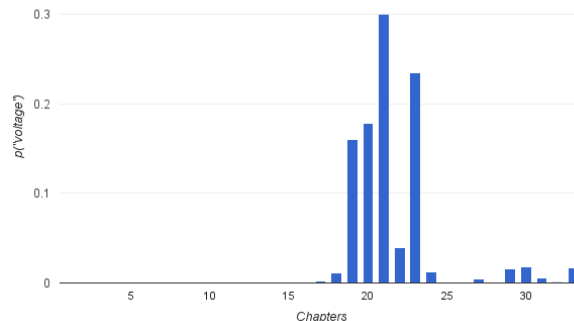

Figure 2: The probability of the term voltage over all chapters. The correct chapter (19) is the first spike, but not the chapter with the highest frequency (21).

Khan Academy Module	Textbook
Physics	*College Physics* by OpenStax
Physics	*Mechanics* by Benjamin Crowell
Biology	*Biology* by OpenStax
Chemistry	*Chemistry* by OpenStax

Table 1: Test materials; the first row was used for training hyperparameters.

as *pickChapterForObj* (see Algorithm 1).

4 Evaluation and Results

All tuning of hyperparameters (tf-idf filtering, the minimum frequency of a learning objective term and the threshold for a spike) was done on augmentation of the Khan Academy physics module with the OpenStax physics textbook. Dataset details appear in Table 1.

For each of the three test modules, we picked a random subset of 10 submodules and split this into two disjoint sets with 5 submodules for each. We recruited four judges and had two judges label each of these disjoint sets, so in total all submodules were labeled twice. For every Khan Academy submodule, the judges were told to select any and all chapters in the textbook that explained the same concepts as that submodule. A fifth judge (one of the authors) broke ties between any discrepancies in answers from the first two judges.

Precision was calculated as $\frac{\sum_{i=0}^{M} N_i}{N}$ where M is the number of submodules, N_i is the number of chapters that were correctly matched for submodule i, and N is the total number of chapters that were

Biology			
Method	Precision	Recall	F1
Tf-idf	**1.0**	0.53	0.69
Term freq	0.88	**0.74**	**0.80**
Spikes	0.87	0.68	0.76
Chemistry			
Method	Precision	Recall	F1
Tf-idf	**1.0**	0.71	**0.83**
Term freq	0.58	**0.79**	0.67
Spikes	0.67	0.71	0.69
Physics			
Method	Precision	Recall	F1
Tf-idf	**0.60**	0.67	**0.63**
Term freq	0.50	0.67	0.57
Spikes	0.46	0.67	0.55

Table 2: The results of the tf-idf (Method 1), Term frequency (Method 2), and Spikes (Method 3) Keywords methods.

Algorithm 1 Augmenting Submodules with Chapters

```
1: procedure AUGMENTSUBMODULE
2:     learningObjectives ← extractLearningObjectives()
3:     relevantChapters = [ ]
4:     for objective in learningObjective do
5:         chapter ← pickChapterForObj(objective)
6:         append chapter to relevantChapters
7:     return relevantChapters
8: procedure PICKCHAPTERFOROBJFREQ(term)
9:     return argmax_i f(term, c_i) for i ∈ {1, 2, . . . , |C|}
10: procedure PICKCHAPTERFOROBJSPIKE(term)
11:     for i ∈ {1, 2, . . . , |C|} do
12:         if s(objective, c_i) > P then return i
13:     return None
```

output. Recall was calculated as $\frac{\sum_{i=0}^{M} N_i}{K}$ where K is the total number of gold-standard chapter annotations for the entire module. F1 was the harmonic mean of the precision and recall scores.

The results for the three methods are shown in Table 2. The tf-idf document similarity method (Method 1) achieves high precision, but lower recall because it only selects one chapter per module. Surprisingly the spikes method (Method 3) performed worse than the term frequency method (Method 2). We believe that this is because there were few occasions in the test set where the chapter with the highest frequency of a term did not correspond to the chapter that a term was explained in.

5 Limitations

Textbooks that are organized differently from the Khan Academy module are more difficult to attain good results on. For example, our methods get much lower results on the physics module because the physics textbook used does not cover certain topics in the Khan Academy physics module, and our methods do not recognize when a term is not being taught.

In addition, the frequency and spike methods have trouble recognizing where a term is explained for terms such as "force" that occur with high frequency throughout a textbook. Both methods also make the simplifying assumption that the learning objectives

for a module can be represented by keyphrases extracted from submodule titles.

6 Conclusions and Future Work

We have presented three simple methods for augmenting Khan Academy modules with textbook chapters. The tf-idf method achieves high precision but lower recall, so we also showcased two methods (term frequency and spikes) that extract learning objectives and attempt to determine which chapters the learning objectives are located in. These results show great promise for using textbooks to automatically improve online learning materials developed for other purposes.

However, so far we have only evaluated our methods in science domains. Our methods may work less well in other domains where the important terms are less technical, and learning objectives cannot be as well represented by such terms.

In addition, for this work, it was known in advance which textbooks were to be aligned to a module. In a more realistic setting, the application must first select an appropriate textbook for the module, perhaps based on both the subject of the textbook and its level of complexity.

Lastly, our current work provides a coarse augmentation by showing entire relevant chapters to the learner; a useful next step will be to extract relevant excerpts from the chapters.

References

Rakesh Agrawal, Sreenivas Gollapudi, Krishnaram Kenthapadi, Nitish Srivastava, and Raja Velu. 2010. Enriching textbooks through data mining. In *Proceedings of the First ACM Symposium on Computing for Development*, page 19. ACM.

Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. 2011. Identifying enrichment candidates in textbooks. In *Proceedings of the 20th international conference companion on World wide web*, pages 483–492. ACM.

Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. 2013. Studying from electronic textbooks. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1715–1720. ACM.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 2010. *Modern information retrieval, 2nd Edition*. Addison Wesley.

Danish Contractor, Kashyap Popat, Shajith Ikbal, Sumit Negi, Bikram Sengupta, and Mukesh K Mohania. 2015. Labeling educational content with academic learning standards. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, April.

Ditty Mathew, Dhivya Eswaran, and Sutanu Chakraborti. 2015. Towards creating pedagogic views from encyclopedic resources. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 190–195.

Rebecca Pitt. 2015. Mainstreaming open textbooks: Educator perspectives on the impact of openstax college open textbooks. *The International Review of Research in Open And Distributed Learning*, 16(4).