Data Mining for Improving Textbooks

Rakesh Agrawal Sreenivas Gollapudi Anitha Kannan Krishnaram Kenthapadi
Search Labs, Microsoft Research
Mountain View, CA, USA
{rakesha, sreenig, ankannan, krisken}@microsoft.com

ABSTRACT

We present our early explorations into developing a data mining based approach for enhancing the quality of textbooks. We describe a diagnostic tool to algorithmically identify deficient sections in textbooks. We also discuss techniques for algorithmically augmenting textbook sections with links to selective content mined from the Web. Our evaluation, employing widely-used textbooks from India, indicates that developing technological approaches to help improve textbooks holds promise.

1. INTRODUCTION

Education is known to be a key determinant of economic growth and prosperity [49; 25]. While the issues in devising a high-quality educational system are multi-faceted and complex, textbooks form one type of educational input most consistently associated with gains in student learning [46]. They are the primary conduits for delivering content knowledge to the students and the teachers base their lesson plans mainly on the material given in textbooks [21].

Considerable research has gone into investigating what makes for good textbooks [24; 30; 48]. There has also been work on designing ideal textbooks [9; 32; 43]. While several factors determine the quality of a textbook, there is general agreement that the good textbooks should present concepts in a coherent manner and provide adequate coverage of important concepts.

Unfortunately, many textbooks, particularly from emerging regions, suffer from two major problems: (1) the lack of clarity of language and incoherent presentation of concepts, and (2) inadequacy of information provided [1]. We quote from a critique of a grade IX Indian History textbook [36]: "The whole (medieval) period has been presented as a dull and dry history of dynasties, cluttered with the names and military conquests of kings, followed by brief acknowledgements of 'social and cultural life', 'art and architecture', 'revenue administration', and so on. The entire Mughal period (1526-1707) is disposed of in six pages."

In order to address the first problem, we present a diagnostic tool for algorithmically identifying those sections of a textbook that are not well-written and hence can benefit from rewriting. The tool uses a probabilistic decision model, which is based on the notion of the dispersion of key concepts occurring in the section and the syntactic complexity of writing [5].

To address the second problem, we draw upon the learning research that shows that the linking of encyclopedic information to educational material can improve both the quality of the knowledge acquired and the time needed to obtain such knowledge [13]. It is also shown that the use of visual material enhances learning, not only by enabling retention of information but also by promoting comprehension and transfer [11; 39]. We, therefore, present techniques for algorithmically augmenting textbook sections with links to selective articles and images mined from the Web. For this purpose, we identify key concept phrases occurring in a section, which are then used to find web articles representing the central concepts presented in the section [6]. Using them, we also mine web images most relevant to a section, while respecting the constraint that the same image is not repeated in different sections of the same chapter [4].

We have applied the proposed techniques to high school textbooks published by the Indian National Council of Educational Research and Training (NCERT). The preliminary results are encouraging and indicate that developing technological approaches to improving textbooks is a promising direction for research.

The paper proceeds as follows. We first describe our method for determining key concepts and the relationships between them in §2. We then discuss our methodology for diagnosing deficient sections in §3. Techniques for augmenting sections with authoritative web articles and images are presented in §4 and §5 respectively. Illustrative results from the empirical evaluation of these techniques are presented in §6. Finally, §7 presents conclusions and directions for future work. We have derived this paper from papers published elsewhere [4; 5; 6]. Here, we focus on describing the main ideas and techniques and refer the reader to the original papers for in-depth descriptions.

2. KEY CONCEPTS

The basic building block underlying our approach is the identification of key concepts described in the book and inferring the relationships between them. We discuss it first.

2.1 Determining Key Concepts

If a textbook includes a back-of-the-book index [38], it can be used for obtaining concept phrases. Unfortunately, not all books contain such indices; *e.g.*, in a study reported in [8], only 55% of the 113 books examined included them. Fortu-

Algorithm 1 DETERMINEKEYCONCEPTS

Input: A section of text s; Pattern R for detecting terminological noun phrases; Pruning parameters Θ.
Output: The set of key concept phrases for s.

1: Tag every sentence in s using a POS tagger. (§*2.1.1*)
2: Compute the set C of terminological noun phrases that maximally match the pattern R. (§*2.1.2*)
3: Prune phrases from C whose POS tagging is inconsistent with a lexical database, but the tag cannot be uniquely corrected using the latter. (§*2.1.3*)
4: Prune common phrases from C based on the probability of occurrence of the phrases on the web and Θ. (§*2.1.4*)
5: Return C.

nately, there is rich literature on algorithmically extracting key phrases from a document that can guide the task of extracting key concepts [7; 42; 50].

After studying several textbooks, we devised the following approach (Algorithm 1). Concepts in our system correspond to *terminological noun phrases*. We first form a candidate set of concepts using linguistic patterns, with the help of a part-of-speech tagger. We used two of the linguistic patterns proposed in [31] that have been used widely in the NLP community. We supplemented this set by a third pattern based on our inspection of the key concepts we identified by studying books on different subjects. We then exploit complementary signals from a different source, namely, a lexical database, to correct errors made by the part-of-speech tagger. Next we eliminate both malformed phrases and very common phrases, based on the probabilities of occurrences of these phrases on the Web. The reason for eliminating common phrases is that they would be already well understood.

Our implementation employs the Stanford POS Tagger [45] for part-of-speech tagging, WordNet [18] as the lexical database, and Microsoft Web N-gram Service [47] to aid pruning of malformed and common phrases. Our methodology, however, is oblivious to the specific tools, though the performance of the system is dependent on them. We summarize our approach in Algorithm 1 and discuss each step in detail below.

2.1.1 Part-of-speech Tagging

We tag every sentence in the given text using Stanford POS Tagger. We note that one could also use a shallow parser (*e.g.* [2]) for this task. The tagger assigns a unique part-of-speech to each word in a sentence. It predicts the part-of-speech tag even for an unknown word (such as a proper noun) by exploiting the context of the word in a sentence. The corpus may contain poorly formed sentences, due to pdf parsing issues as well as the presence of text extracted from tables, mathematical equations, and other non-grammatical structures. For such sentences, the assigned part-of-speech tags may be incorrect.

2.1.2 Detecting Terminological Noun Phrases

We next form a candidate set of concepts by determining the terminological noun phrases present in the text. The concepts of interest in our application typically consist of noun phrases containing adjectives, nouns, and sometimes prepositions. It is rare for concepts to contain other parts of speech such as verbs, adverbs, or conjunctions.

We consider three patterns (P_1, P_2, and P_3) for determining terminological noun phrases. The first two of these are from [31] and the third is the one we added. We can express the three patterns using regular expressions as:

$$P_1 = C^*N$$

$$P_2 = (C^*NP)^?(C^*N)$$

$$P_3 = A^*N^+$$

where N refers to a noun, P a preposition, A an adjective, and $C = A|N$. The pattern P_1 corresponds to a sequence of zero or more adjectives or nouns, ending with a noun, while P_2 is a relaxation of P_1 that also permits two such patterns separated by a preposition. Examples of the former include "cumulative distribution function", "fiscal policy", and "electromagnetic radiation". Examples of the latter include "degrees of freedom" and "Kingdom of Asoka". P_3 corresponds to a sequence of zero or more adjectives, followed by one or more nouns. This pattern is a restricted version of P_1, where an adjective occurring between two nouns is not allowed. The motivation for this pattern comes from sentences such as the following: "The experiment with Swadeshi gave Mahatma Gandhi important ideas about using cloth as a symbolic weapon against British rule". As a consequence of allowing arbitrary order of adjectives and nouns, "Mahatma Gandhi important ideas" is detected as a terminological noun phrase by pattern P_1. On the other hand, pattern P_3 would result in the better phrases, "Mahatma Gandhi" and "important ideas".

Our candidate concepts comprise of maximal pattern matches. Thus, we will not have "distribution function" as a candidate in the presence of "cumulative distribution function". The intuition is that it is better to have more specific concepts than general concepts. A similar strategy was used in [33].

It was found in the empirical study reported in [6] that the pattern P_1 outperforms P_2. The pattern P_3 exhibited slightly better performance than P_1 in this study.

2.1.3 Correcting Errors using WordNet

The Stanford POS Tagger can make errors on poorly formed sentences. We experimented with using WordNet to detect these errors and correct them. WordNet is a large lexical database that groups words into sets of cognitive synonyms called synsets, each expressing a distinct concept. We use WordNet to determine possible parts of speech (noun, adjective, verb, adverb) for words in its knowledge base. However, WordNet would fail to recognize words absent in its database. WordNet being a hand curated system should have better accuracy than an automated parsing tool, but lower coverage. We therefore use WordNet as a validation and error-correcting tool.

We check whether the parts of speech tags assigned by the Stanford POS Tagger are consistent with those provided by WordNet. We say that disagreement occurs for a phrase if for some word w in the phrase, (a) WordNet recognizes w and returns one or more part-of-speech tags *and* (b) the part-of-speech tag assigned by the Stanford POS Tagger is *not* among the part-of-speech tags assigned by WordNet. For example, for the phrase "steatite micro beads", the Stanford POS Tagger assignment is `<Adjective><Noun><Noun>`

whereas the WordNet assignment is <Noun><Adjective><Noun>. In such cases, we change the POS Tagger assignment to the WordNet assignment, provided the latter still satisfies the linguistic pattern. In the above example, the assignment will be modified to <Noun><Adjective><Noun>.

However, there may be cases where the WordNet assignment is not unique. For example, for the phrase "control measures", WordNet has a non-unique assignment: <Noun|Verb><Noun|Verb>. Thus, the POS Tagger assignment <Adjective><Noun> is in disagreement with the WordNet assignment, but it cannot be uniquely corrected and hence we drop the phrase from the candidate set.

The empirical evaluation demonstrated that a lexical database such as WordNet can be quite complementary to a generic part-of-speech tagger such as the Stanford POS Tagger, and we were able to successfully use WordNet for correcting errors made by the POS tagger [6].

2.1.4 Pruning using the Web N-gram Service

The set of candidate phrases generated in the previous step is likely to contain a number of common knowledge phrases as well as some malformed or unimportant long phrases. For identifying such phrases, we obtain the probability of occurrence of the phrase on the Web using the Microsoft Web N-gram Service. We use this probability as a proxy for whether the phrase is part of common knowledge, since a common knowledge phrase is likely to have a significant presence on the Web. Similarly this probability can also indicate whether the phrase is malformed, as such phrases are less likely to occur on the Web. Thus, after obtaining the probability scores for each phrase, we compute the score distribution across phrases over the entire corpus, and prune based on this distribution to remove undesirable phrases.

The Microsoft Web N-gram Service provides the probability of occurrence of a given phrase over three corpora: bodies of web pages, titles of pages, and anchor texts for web pages. Compared to title or body, we found that the anchor provided a stronger signal, perhaps because the anchor text represents how other web authors succinctly describe the target page.

Given the distribution D of N-gram log probability scores of candidate phrases, we compute certain parameterized statistical boundaries. Let Q_1 denote the first quartile, that is, Q_1 satisfies $Pr_{x \in D}(x \leq Q_1) = 0.25$. Similarly let Q_3 denote the third quartile, that is, Q_3 satisfies $Pr_{x \in D}(x \leq Q_3) = 0.75$. The interquartile range $IQR = Q_3 - Q_1$ is a measure of mid-spread of the distribution. Given non-negative parameters t_1 and t_2, we can define *fences* on both ends of the distribution:

$$LF(t_1) = Q_1 - t_1 \cdot IQR,$$

$$UF(t_2) = Q_3 + t_2 \cdot IQR.$$

We prune phrases whose scores are not within the fences as the phrases with scores below the lower fence ($LF(t_1)$) are likely to be malformed and those with scores above the upper fence ($UF(t_2)$) are likely to be of common knowledge. As the distribution of scores is not symmetric around the mean, we may need to select different pruning parameters.

Our empirical evaluation showed that our approach was quite effective in identifying and pruning concepts that were malformed or represented common knowledge concepts [6].

Algorithm 2 DETERMINECONCEPTGRAPH

Input: The set of key concept phrases C for a given section s; An authoritative structured external source of concepts that also contains relationships between them (e.g. Wikipedia).

Output: The concept graph for s.

1: Determine the set V of nodes corresponding to concepts in C that match an article title from the external source.
2: Let W denote the set of all links in the external source. Define $E = \{(v_1, v_2) | v_1, v_2 \in V \wedge (v_1, v_2) \in W\}$. Compute the directed graph $G = (V, E)$ thus induced by the links in W.
3: Return G.

2.2 Concept Graph

Having determined the set of concepts, a straightforward approach to derive relationships between concepts would be to manually label the concept pairs. However, labeling is a laborious and subjective task. We instead consider an authoritative structured external source of concepts that also contains relationships between the concepts and use it to infer relationships between the textbook concepts (Algorithm 2). Our implementation maps textbook concepts to Wikipedia articles and treats a concept c_1 to be related to another concept c_2 if the Wikipedia article corresponding to c_1 has a link to the Wikipedia article corresponding to c_2. We only consider concept phrases that match the title of a Wikipedia article exactly. If any Wikipedia article is redirected to another article, we follow the redirect link till an article is found. We then consider the directed graph induced by these mapping articles and the Wikipedia links between them, thereby obtaining a concept graph that encapsulates the relationships between the concepts.

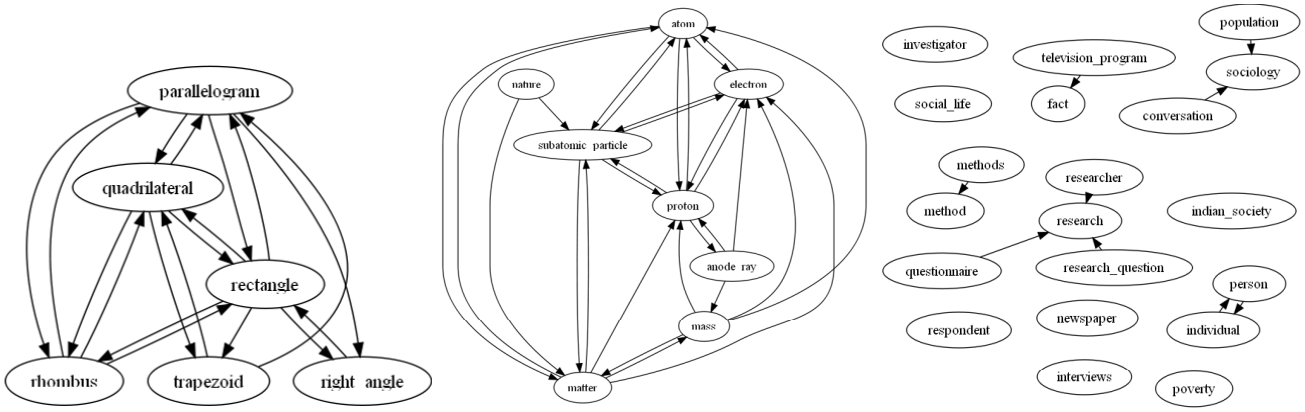
3. DIAGNOSING DEFICIENT SECTIONS

Our decision model for identifying a poorly written section is based on the dispersion of key concepts mentioned in the section and the syntactic complexity of the writing. The model requires a tune set for learning its parameters. While human judgments may seem like an obvious way to obtain a tune set, it is difficult to assemble a sufficiently large group of qualified judges who can provide consistent ratings. Hence we generate the tune set automatically in a novel way. This procedure maps sampled text book sections to the closest versions of Wikipedia articles having similar content and uses the maturity of those versions to assign need-for-exposition labels. The maturity of a version is computed by considering the revision history of the corresponding Wikipedia article and convolving the changes in size with a smoothing filter. We first discuss the rationale for choosing these decision variables and formally define them, followed by a discussion of the model and the generation of the tune set.

3.1 Decision Variables

3.1.1 Dispersion

After going through several textbooks, we observed that a section that discussed concepts related to each other was more comprehensible than one that discussed many unre-



(a) A section with very low dispersion (Grade IX Mathematics book)

(b) A section with low dispersion (Grade IX Science book)

(c) A section with very large dispersion (Grade XII Sociology book)

Figure 1: Concept graphs illustrating dispersion

Algorithm 3 ComputeDispersion

Input: A textbook section s.
Output: Dispersion value for section s.

1: Compute the set of concepts C present in s. (§2.1)
2: Infer the concept graph E for the concepts in C. (§2.2)
3: $dispersion(s) := 1 - \frac{|E|}{|V|(|V|-1)}$.

lated concepts. We formally capture this intuition by defining a measure of dispersion over key concepts.

Let V represent the set of key concepts in a section s. Let rel be a binary relation that determines whether a concept in V is related to another concept in V, that is, $rel(x, y)$ is *true* if concept x is related to concept y and *false* otherwise. We define *dispersion* of a section as the fraction of ordered key concept pairs that are not related:

$$dispersion(s) := \frac{|\{(x,y)|x,y \in V \wedge x \neq y \wedge \neg rel(x,y)\}|}{|V|(|V|-1)}. \quad (1)$$

We note that *dispersion* takes values between 0 and 1, with 0 corresponding to a section where all key concepts are mutually related and 1 corresponding to a section with mutually unrelated key concepts.

Algorithm 3 describes the computation of dispersion for a given section. We first identify concepts following the method discussed in §2.1, employing the pattern A^*N^+, where A is an adjective and N a noun. We then obtain the concept graph of relationships as described in §2.2, with isolated nodes removed. We note that the dispersion as defined in Eq. 1 is the same as 1 minus the edge density of this resulting graph, which we compute in the last step of Algorithm 3.

We illustrate our notion of dispersion through some examples from the NCERT textbooks. Figure 1(a) and 1(b) show the concept graphs for two sections with small dispersion. The first section titled "Types of Quadrilaterals" from the Grade IX Mathematics book has 19 directed edges over 6 nodes with dispersion 0.37 and the second section titled "Charged Particles in Matter" from the Grade IX Science book has 29 directed edges over 8 nodes with dispersion 0.48. Indeed the concepts within each of these sections are quite related to each other, resulting in low dispersion values. Figure 1(c) shows the concept graph for a section with large dispersion (with some isolated nodes also shown). This section titled "Variety of Methods" from Grade XII Sociology book has 9 edges over 13 non-isolated nodes, contributing to a dispersion value of 0.94. The section discusses rather unrelated concepts, leading to large dispersion.

3.1.2 Syntactic Complexity

To measure syntactic complexity of writing, our first instinct was to use readability formulas [16]. Table 1 summarizes some of the popular ones and the variables they use. We observe that all formulas base their calculations on two classes of variables. First, they all use a sentence structure measure, generally sentence length, the underlying intuition being that longer sentences are harder to read and comprehend. The sentence length can be in terms of the number of letters or the number of words, though the empirical evidence from past studies overwhelmingly favors the number of words. The second measure they use captures the difficulty of the vocabulary at word level in terms of word familiarity or word length. The Dale long list [14] is frequently used for computing word familiarity. We do not employ word familiarity because of potential vocabulary mismatch between textbooks written in local variants of English and the Dale list. The word length can be defined in terms of the number of syllables or the number of letters. Both the Coleman-Liau Index and the Automated Readability Index calculate word lengths as the number of letters. Their primary consideration, however, is data processing efficiency and the effectiveness of this approach is suspicious [16]. Another approach is to compute word length in terms of the number of syllables, the intuition being that words with more syllables are more complex.

We also note that different readability formulas combine the above two measures differently and the combinations are learned with respect to specific datasets (often McCall-Crabbs Standard Test Lessons in Reading [35]). As a result, these formulas are highly correlated, a fact we confirmed in our experiments. We find it unnatural to directly use the readability scores determined by these formulas as variables in the decision model.

Flesch Reading Ease Score	206.835	−	84.6	×	S/W	−	1.015	×	W/T
Flesch-Kincaid Grade Level	−15.59	+	11.8	×	S/W	+	0.39	×	W/T
Dale-Chall Grade Level	14.862	−	11.42	×	D/W	+	0.0512	×	W/T
Gunning Fog Index			40	×	C/W	+	0.4	×	W/T
SMOG Index	3.0	+	$\sqrt{30}$	×	$\sqrt{C/T}$				
Coleman-Liau Index	−15.8	+	5.88	×	L/W	−	29.59	×	T/W
Automated Readability Index	−21.43	+	4.71	×	L/W	+	0.50	×	W/T

C	=	Number of words with three syllables or more
D	=	Number of words on the Dale Long List
L	=	Number of letters
S	=	Number of syllables
T	=	Number of sentences
W	=	Number of words

Table 1: Popular readability formulas and their variables

After considerable experimentation, we settled on the following two variables as measures for the syntactic complexity of writing:

1. *Average sentence length*: average number of words per sentence in the section.

2. *Average word length*: average number of syllables per word in the section.

See [12; 17] for algorithms for computing the number of syllables per word. The number of syllables in a word can also be approximated by counting consonant-separated vowels. Each group of adjacent vowels counts as one syllable (for example, 'ea' in 'real' contributes one syllable, whereas 'e...a' in 'regal' contributes two syllables), but an 'e' occurring at the end of a word does not contribute to syllable count. Each word has at least one syllable.

3.2 Decision Model

We take a learning approach to arrive at the model for deciding whether a book section can benefit from rewriting. Our proposed model is probabilistic and its parameters are learned using an algorithmically generated tune set. The tune set consists of sections with different maturity, the intuition being that the more immature a section, the greater the need for its revision.

3.2.1 Model

Our goal is to learn a decision model that can provide a probabilistic score of whether a textbook section requires revision based on the values of decision variables for that section. We would also like such a decision model to automatically learn the relative importance between the decision variables. The binary logistic regression eminently lends itself to this desiderata.

Let $\mathbf{z}$ represent a section's decision variables: a three dimensional vector whose components are the average sentence length, average word length, and dispersion. Given $\mathbf{z}$, the binary logistic regression predicts the probabilistic score that a section needs revision (i.e., label $y = 1$) through the logistic function:

$$P(y = 1|\mathbf{z}, \mathbf{w}) = \frac{1}{1 + \exp\{-(b + \mathbf{z}^T\mathbf{w})\}}.$$

The parameter $\mathbf{w}$ is the weight vector of the function, with each component w_j measuring the relative importance of the decision variable z_j for predicting the label y.

The weight vector $\mathbf{w}$ is learned from a tune set consisting of N textbook sections: $\{\mathbf{Z}, \mathbf{y}\} = \{(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_N, y_N)\}$, with $(\mathbf{z}_i, y_i)$ representing the decision variable vector $\mathbf{z}_i$ and the label y_i for the i^{th} textbook section. The optimal $\mathbf{w}$ is the one that maximizes the conditional log-likelihood of the labels in the tune set:

$$\arg\max_{\mathbf{w}} \log P(\mathbf{y}|\mathbf{Z}, \mathbf{w}) = \arg\max_{\mathbf{w}} \sum_{i=1}^{N} \log P(y_i|\mathbf{z}_i, \mathbf{w}).$$

Algorithm 4 GENERATETUNESET

Input: A collection of sections from a textbook corpus; A collection of versioned documents from an authoritative web resource such as Wikipedia; Threshold parameters θ_1 and θ_2.

Output: A tune set consisting of a subset of sections, each labeled either 1 (Revise) or 0 (Don't).

1: **for** each section s **do**
2: Map section s to a set $W(s)$ of most similar versioned documents from the web resource, along with their similarity scores $sim(s, v)$ $\forall v \in W(s)$. (§*3.2.3*)
3: Compute immaturity score $\tilde{m}(v)$ for each versioned document $v \in W(s)$. (§*3.2.4*)
4: Compute immaturity score $m(s)$ for section s by aggregating immaturity scores $\tilde{m}(v)$ for $v \in W(s)$, weighted by their similarity scores $sim(s, v)$.
5: $Label(s) := 1$ if $m(s) > \theta_1$; 0 if $m(s) < \theta_2$; undefined otherwise.
6: Output $\langle s, Label(s)\rangle$ for sections s where $Label(s)$ is either 1 or 0.

3.2.2 Generating Tune Set

Given the difficulty of obtaining manual judgments, we propose using meta data associated with textbooks to obtain labels. One such meta data is the immaturity level of a section; an immature section hinders the positive learning experience of a student, and therefore calls for revision. However, immaturity computation requires access to rich data such as extent and timing of the revisions, which is typically not available for textbooks. We, therefore, resort to an indirect device for estimating the maturity of a section.

We note that authoritative information resources on the Web, such as Wikipedia, are created through collective efforts of multiple authors. The content gets repeatedly updated until writers expressing opinions on the subject come to a consensus. As new information becomes available, this cycle of revisions repeats. A key aspect of such web resources is that the revisions are recorded and maintained by the source. Hence, we map a textbook section to the most similar version of a similar article in a web resource and use the immaturity of that version as the proxy for the immaturity of the textbook section.

The tune set generation is outlined in Algorithm 4. We sample a subset of textbook sections across all subjects and

classes. For each section, we find a small set of closest matching versions in the web resource that are similar in content. The matches are found using the technique described in §*3.2.3*. We then compute the immaturity for these versions using the technique given in §*3.2.4*. The immaturity scores are then aggregated through a weighted combination (weights are the normalized similarity scores) to produce the maturity score for the textbook section. This score is then converted into a decision on what label should be assigned to this book section. Note that we need only a small amount of labeled data since the model has a very small number of parameters.

We observe that the immaturity computation is reliable only at the extreme ends: very high values or very low values of scores. The parameters θ_1 and θ_2 allow us to achieve this goal. Their values are empirically determined, balancing the need for high precision with the need for having sufficient labeled data.

3.2.3 Computing similarity

In a document model where each document is treated as a bag (multi-set) of words, a well-known measure of similarity between documents A and B is the Jaccard index, defined as $sim(A, B) := |A \cap B|/|A \cup B|$. Here, we note that the terms and their associated weight, i.e., importance (e.g., *tf-idf*) in the document gives raise to the multi-set representation of the document. Thus $A = \{< x_1, w_{x_1} >, < x_2, w_{x_2} >, \ldots, < x_n, w_{x_n} >\}$. The large and often varying sizes of documents (i.e., cardinality of the sets) and further, terms with varying weights can make this similarity computation expensive. We use the well-known min-wise independent permutations [10] to get around these problems.

We proceed as follows. Given a document A, we convert it to a set $\tilde{A} = \{x \in A | x \in A \bigwedge R(x) \leq w_x\}$ using a consistent hash function $R(x)$ that maps words in the document uniformly and randomly in the interval $[0, 1]$. In other words, we include the significant terms in the document in the newly defined document set. Next, we compute the min-wise independent permutation of $\tilde{A}$ as $MH(\tilde{A}) := \arg\min_x \{R(x) | \ x \in \tilde{A}\}$. Thus, $MH(\tilde{A})$ denotes the leftmost element of $\tilde{A}$ in the permutation. Now, for any two documents A and B, $|A \cap B|/|A \cup B| \approx \Pr[MH(\tilde{A}) = MH(\tilde{B})]$ [23]. Finally, we compute H min-hashes to yield the *sketch* of A, $S(A) = \{MH_1(\tilde{A}), MH_2(\tilde{A}), \ldots, MH_H(\tilde{A})\}$. Repeat for B. Now, $|S(A) \cap S(B)|/|S(A) \cup S(B)|$ gives the estimate for $sim(A, B)$.

3.2.4 Computing immaturity

Consider a web repository in which a new version of a document is created at the end of the day, ignoring multiple updates to the document within a day. Older versions of a document are saved when a new version is created. We observe that paraphrasing, additions or deletions indicate the amount of revision. Thus, the relative change in the size of the document is an indicator of the maturity of a version (the smaller the change, the higher the maturity). The number of days for which a version remains the latest version is also an indicator of the maturity of the version (the longer the duration, the higher the maturity). Finally, people tend to consult nearby versions when creating a revision. Thus, maturity is a local phenomenon driven by local context. Armed with these observations, we proceed as follows.

Assume days are numbered from 1 to the current day T.

Algorithm 5 AUGMENTWITHARTICLES

Input: A textbook section s.
Output: An ordered list of links to top k articles for embellishing section s.

1: Compute the set C of concepts present in s. (§2.1)
2: Infer the concept graph E for the concepts in C. (§2.2)
3: Compute k. (Eq. 2)
4: Compute the authority score of nodes in E.
5: Return the top k nodes in the decreasing order of node authority score, excluding any node corresponding to a concept embellished earlier in the textbook.

Consider a document whose initial version v_1 was created on day 1. Let $\mathbf{L}$ be a vector of length T whose i^{th} component L_i is equal to the size of the document (in number of words) on day i. Define a vector $\delta(\mathbf{L})$ whose i^{th} component is the relative change in document size between neighboring days i and $i-1$:

$$\delta(L)_i = |L_i - L_{i-1}|/L_i.$$

For a particular version v created on day d, we define its *immaturity* $\tilde{m}(v)$ to be the value of convolution between $\delta(L)$ and a smooth filter $\mathbf{h}$ on day d:

$$\tilde{m}(v) := (\delta(L) * \mathbf{h})_d = \sum_{j=\max(-K/2,1-d)}^{\min(K/2,T-d)} h_j \delta(L)_{d+j},$$

where K is a parameter of the filter used in the convolution. The convolution with a smooth filter allows for modeling immaturity as a smooth continuous process, and the use of local neighborhood enables incorporating local context. We employ the frequently used Hann Filter

$$h_j = 0.5(1 + \cos(2\pi j/K))$$

that has K days spatial support with a smooth fall off in the chosen K sized neighborhood.

We note that there have been efforts (*e.g.* [26]) to assign quality index to Wikipedia articles taking into account edit history of the article such as the frequency and size of edits and the type and reputation of the authors. However, we are not aware of any work targeted at computing the maturity of an arbitrary version of a Wikipedia article, and the technique we presented could be of independent interest.

4. AUGMENTING WITH AUTHORITATIVE ARTICLES

Our goal is to embellish a textbook section with links to authoritative articles most relevant to the central concepts discussed in the section. Our implementation uses Wikipedia as the source of supplementary material. In order to contain the cognitive burden on the reader, we add only up to k links.

As described in Algorithm 5, we first identify key concept phrases present in the section. We use the algorithm from §2.1 for this purpose. We next form the concept graph for the section, inferring relationships between the concepts thus identified, using the algorithm from §2.2. We now compute the authority score (*e.g.* page rank) of the nodes in the concept graph. We then sort the nodes in the decreasing order of their authority scores, select top k nodes, and augment

the section with links to articles corresponding to them. The intuition is that the central concepts present in a section will be related to many concepts mentioned in the section. Further, given the progressive learning nature of the textbooks, it is worthwhile to exclude concepts that have already been used for augmentation earlier in the textbook.

The number of articles, k, selected for embellishing a section can be determined using the distribution of the node authority scores. The node authority scores appear to follow Zipf's ranked distribution, $X_r \propto r^{-1/\alpha}$, where X_r is the value of the r^{th} ranked node authority score and α is the tail index parameter of the underlying Pareto distribution. The tail index can be estimated by regressing the log of order statistics on the log of the scores. For a desired coverage, c (say, 80%) and a limit k_0 (say, 3) on the maximum number of concepts to be shown, we obtain

$$k = \min\left(k_0, \left(c + (1-c)n^{\left(\frac{1}{\alpha}-1\right)}\right)^{\frac{\alpha}{\alpha-1}} \cdot n\right), \qquad (2)$$

where n is the number of nodes in the induced graph. While this determination can be made empirically as well, we propose fitting Zipf's distribution to the node authority scores as it will help to characterize the distributions over different subjects and over different grade levels, with varying α values.

5. AUGMENTING WITH AUTHORITATIVE IMAGES

Our goal is to find a small number of images that are most relevant to enhance the understanding of a particular section of the textbook, shunning repetition of an image in different sections of the same chapter. Our solution has three components:

Image Mining. This component comprises of algorithms that mine the web for images relevant to a particular section and provide a ranked list of top k images along with their relevance scores. It is preferable to have algorithms that make use of orthogonal signals in their search for images in order to have a broad selection of images to choose from. We provide two specific algorithms, namely COMITY and AFFINITY, which satisfy these properties.

Image Assignment. The image mining algorithms provide locally optimal solution in that they yield images that are best suited for the given section. Consequently, the same image might be selected for different sections of a chapter, giving rise to the need for chapter level optimal assignment of images.

Given a set of candidate images and their relevance scores for every section of a chapter, the image assignment component assigns images to various sections in a way that maximizes the relevance score for the chapter while maintaining the constraints that no section has been assigned more than a certain maximum number of images and no image is used more than once in the chapter. We provide a polynomial time algorithm for implementing the optimizer.

Image Ensembling. Since the relevance scores provided by different image mining algorithms will in general be incomparable, the assignment of images to different sections of a chapter needs to be performed separately for each algorithm. The image ensembling component aggregates the ranked lists of image assignments to produce the final result.

Algorithm 6 COMITY

Input: A textbook section j; Number of desired image results k; Number of desired image search results per query t; Number of desired concept phrases c.

Output: A list of top k image results from web, along with relevance scores.

1: Obtain c concept phrases from section j. (§2.1)
2: Form queries consisting of two and three concepts phrases each ($\binom{c}{2} + \binom{c}{3}$ queries in total).
3: Obtain top t image search results for each of the queries from e different search engines.
4: Aggregate over (potentially $e(\binom{c}{2} + \binom{c}{3})$) lists of images, to obtain λ_{ij} values for each image.
5: Return top k images along with their λ_{ij} values.

While it is possible to use any rank aggregation algorithm, we wanted a voting scheme that considers all the elements of a ranked list and provides consensus ranking. The popular Borda's method fits the bill [41]. Ensembling is done sequentially within a chapter, starting from the first section. Top images selected for a section are eliminated from the pool of available images for the remaining sections. The image assignment is then rerun, followed by an ensembling for the next section.

We first present two algorithms for mining relevant images from the web, followed by the image assignment component consisting of an optimization problem, and finally the image ensembling component.

5.1 Image Mining

Here we give particulars of the COMITY and AFFINITY algorithms, the two algorithms used for obtaining the ranked list of top k images along with their relevance scores for a given section. Note that our system design admits various possible variants of these algorithms as well as additional image mining algorithms one could conceive.

Algorithm COMITY

One might think that one could simply use the text string of a section to query a commercial image search engine and obtain the relevant images. However, the current search engines do not perform well with long queries [27]. Indeed, when we queried the search engines using even the first paragraph of a section, we got none or meaningless results. In one major stream of research on information retrieval with long queries, the focus is on selecting a subset of the query, while in another it is on weighting the terms of the query [51]. This body of research however is not designed to work for queries consisting of arbitrary textbook sections.

Algorithm 6 (COMITY) is based on using the key concepts present in a section to query the commercial image search engines. However, each concept phrase in isolation may not be representative of the section as a typical book section can discuss multiple concepts. Hence we form $\binom{c}{2} + \binom{c}{3}$ image search queries by combining two and three concept phrases each, in order to provide more context about the section. A relevant image for the section is likely to occur among the top results for many such queries. Thus, by aggregating the image result lists over all the combination queries, we end up boosting the relevance scores of very relevant images for the section. We further increase the coverage by obtaining

Algorithm 7 AFFINITY

Input: A textbook section j; Number of desired image results k; Number of desired closest articles from an authoritative external source t'; Number of desired concept phrases c.

Output: A list of top k image results from the authoritative source, along with value scores.

1: Obtain c concept phrases from section j. (§2.1)
2: Obtain top t' closest articles from the authoritative external source, based on content similarity with section j. (§*3.2.3*)
3: Extract the set of images present in these t' articles, as well as the metadata associated with each image aggregated over all occurrences of the image.
4: For each image i, let n_{ij} := Number of articles in which image i appears, d_{ij} := Number of concept phrases contained in the metadata for image i, w_{ij} := Number of matching words from all concepts in the metadata for image i.
5: Assign the relevance score $\lambda_{ij} := n_{ij}^{w_1} \cdot d_{ij}^{w_2} \cdot w_{ij}^{w_3}$ for image i (w_1, w_2, w_3 determine the relative weight given to the three counts above).
6: Return top k images along with their λ_{ij} values.

and merging results across e different search engines. We treat each search engine as a blackbox [27], that is, we have access to the ranking of results but do not have access to the internals of the search engine such as the score given to a document with respect to a query.

Aggregation across multiple lists is performed as follows. Each of t images in a result list is assigned a position-discounted score equal to $1/(p+\theta)$ where p denotes the position and θ is a smoothing parameter. For the same image occurring in multiple lists, the scores are added, weighted by a function f of the importance of the concept phrase present in the underlying query: $\lambda_{ij} := \sum_q f(\text{Importance scores of concept phrases used in } q) \times (1/(p(i,q,R(q))+\theta))$. Here the summation is over $e(\binom{c}{2} + \binom{c}{3})$ queries issued and $p(i,q,R(q))$ denotes the position of image i in the result list $R(q)$ for query q if i is present in $R(q)$ and ∞ otherwise. This choice is based on our empirical observation that an image occurring among the top results for multiple queries was more relevant to the section than an image that occurred among the top results for only one query.

Algorithm AFFINITY

The intuition behind this algorithm is the observation that the images included in an authoritative article relevant to a topic are often illustrative of the key concepts underlying the topic. We therefore find authoritative articles whose contents have high textual similarity with a given section of the book. We then extract images contained in these articles and use their relevance scores to find top k images for the section.

Algorithm 7 (AFFINITY) first obtains the key concept phrases present in a section as well as the closest authoritative articles from the web. Thus the key topics discussed in the section are available in the form of the concept phrases while the search space for images is refined to the set of articles with high document similarity to the section. The relevance score for an image is computed by analyzing the overlap between the concept phrases and the cumulative metadata associated with the various copies of the image present in the narrowed set of articles. The metadata for an image comprises of text adjacent to the image including caption and alternative text, filename of the image, anchor texts pointing to the image, and queries that led to clicks on the image. The scoring has desirable properties such as: (a) an image occurring in multiple articles gets a higher score, (b) an image whose metadata contains multiple concept phrases gets a higher score, and (c) an image whose metadata contains words from many concepts gets a higher score.

5.2 Image Assignment

Given a set of candidate images relevant to the various sections of a chapter and their relevance scores, the goal of the image assignment component is to allocate to each section the most relevant images, while respecting the constraints that each section is not augmented with too many images and that each image is used no more than once in a chapter. The rationale for these constraints is that an augmentation of a section with too many images will put undue cognitive burden on the reader while the repetition of an image across sections in the same chapter would be redundant for the reader.

First, a few notations. Let $I = \{1, 2, \ldots, n\}$ denote the set of images and $S = \{1, 2, \ldots, m\}$ denote the set of sections in a chapter. Let λ_{ij} denote the (non-negative) relevance score of image $i \in I$ for section $j \in S$ ($\lambda_{ij} = 0$ if the image i is not present in the candidate set of images for section j). Let K_j denote the maximum number of images that can be associated with section j. K_j could be either a fixed integer for all sections or a function of the length of the section j.

This problem admits a natural greedy algorithm. Sort the λ_{ij} values in decreasing order and go through them. At each step, the greedy algorithm picks the highest λ_{ij} value such that an image can still be assigned to section j (that is, less than K_j images have so far been assigned to j) and then assigns image i to section j. This process ends when either all sections have been assigned the maximum number of images or there are no more images to be assigned.

At a first glance, the greedy algorithm might seem optimal in terms of the sum of relevance scores of all assigned images. But the following counterexample shows that the optimal value can be substantially larger. Consider a chapter consisting of two sections and suppose that we want two images each for a section ($K_1 = K_2 = 2$). Represent by (i, λ) that image i's relevance score is λ. Let the top images and their relevance scores obtained by an image mining algorithm for various sections be as follows: $s_1 \leftarrow \langle (i_1, 1), (i_2, 1-\epsilon), (i_3, 1-3\epsilon) \rangle$, $s_2 \leftarrow \langle (i_2, 1-2\epsilon), (i_4, \epsilon), (i_5, \epsilon) \rangle$, where $\epsilon = 0.01$. Then the greedy assignment would be $s_1 \leftarrow \langle i_1, i_2 \rangle$, $s_2 \leftarrow \langle i_4, i_5 \rangle$ with a total score of $2 + \epsilon$. On the other hand, an optimal assignment is $s_1 \leftarrow \langle i_1, i_3 \rangle$, $s_2 \leftarrow \langle i_2, i_4 \rangle$ with a total score of $3 - 4\epsilon$.

We, therefore, instantiate the image assignment component as an optimization problem. We show that this optimization problem can be solved optimally in polynomial time and provide an efficient algorithm as part of the proof. The following is the statement of the optimization problem:

MaxRelevantImageAssignment

$$\max \sum_{i \in I} \sum_{j \in S} x_{ij} \cdot \lambda_{ij} \quad (3)$$

s.t.

$$x_{ij} \in \{0,1\} \quad \forall i \in I \; \forall j \in S \quad (4)$$

$$\sum_{i \in I} x_{ij} \leq K_j \quad \forall j \in S \quad (5)$$

$$\sum_{j \in S} x_{ij} \leq 1 \quad \forall i \in I \quad (6)$$

Here, x_{ij} is an indicator variable that takes value 1 if image i is selected for section j and 0 otherwise. Eq. 4 captures this binary constraint. Eq. 5 ensures that the number of images assigned to a section is at most K_j. Eq. 6 enforces that each image is assigned to at most one section in a chapter. The optimization objective (Eq. 3) is the total relevance score for the chapter, defined as the sum over all sections of relevance scores of the images assigned to the section. Thus the goal of the optimization is to compute the binary variables x_{ij} such that the total relevance score for the chapter is maximized.

Theorem 5.1. MaxRelevantImageAssignment *can be solved optimally in polynomial time.*

Proof. The proof follows by showing an efficient reduction from MaxRelevantImageAssignment to the Maximum Weighted Bipartite Matching problem [40], which admits an efficient polynomial time solution. Given an instance of MaxRelevantImageAssignment, form a complete weighted bipartite graph $G = (V, E)$ as follows. Associate a node u_i with each image $i \in I$ and associate K_j nodes, $v_{j1}, v_{j2}, \ldots, v_{jK_j}$, with each section j. Create an edge between every image node and every section node copy. Weight of the edge (u_i, v_{jk}) is set to λ_{ij} for each $k \in \{1, 2, \ldots, K_j\}$, that is, each of the K_j edges joining an image i to the section j has the same weight, equal to the corresponding relevance score.

We observe that any feasible solution to MaxRelevantImageAssignment corresponds to selecting a matching in G. Given a satisfying assignment of x_{ij}'s, we can obtain a matching in G by picking one of the K_j edges corresponding to any x_{ij} that is set to 1. Similarly, given any matching in G, there is a corresponding feasible solution. Further the objective of MaxRelevantImageAssignment can be maximized by obtaining the maximum weight bipartite matching in G. As the Maximum Weighted Bipartite Matching problem can be solved optimally in $O(nm(n+m))$ time, it follows that MaxRelevantImageAssignment can also be solved optimally in $O(nm(n+m))$ time. □

5.3 Image Ensembling

We next describe our ensembling algorithm for combining the different image assignments. Since the relevance scores computed by the image mining algorithms will be incomparable in general, we combine the results *after* the MaxRelevantImageAssignment optimization has been performed independently for each algorithm. We use only the ordering returned by these algorithms and do rank aggregation without considering the magnitudes of the scores.

We employ Borda's method to merge l ranked lists corresponding to l different image mining algorithms. Borda's method tries to achieve a consensus ranking and satisfies certain desirable properties such as reversal symmetry [41]. It assigns a score corresponding to the positions in which an image appears within each ranked list of preferences, and the images are sorted by their total score.

Algorithm 8 Ensemble

Input: Set of sections $S = \{1, 2, \ldots, m\}$ in a textbook chapter; Set of images $I = \{1, 2, \ldots, n\}$; Number of desired images K_j for each section $j \in S$; Scores assigned by l different image mining algorithms for each image $i \in I$; Orderings produced after the optimization for these l algorithms.

Output: A new assignment of images to sections.

1: Let $I_0 := I$ and $S_0 := S$. For each of l image mining algorithms, perform MaxRelevantImageAssignment optimization over I and S to get an assignment of images for all sections in S.
2: **for** section $j = 1$ to m **do**
3: Merge l ranked lists (corresponding to l algorithms) for section j using Borda's method, and assign the top K_j images from the merged list to section j. Let A_j denote the set of assigned images.
4: Remove the assigned images from consideration for subsequent sections, that is, $I_j := I_{j-1} \setminus A_j$ and $S_j := S_{j-1} \setminus \{j\}$.
5: For each of l image mining algorithms, perform MaxRelevantImageAssignment optimization over I_j as the set of images and S_j as the set of sections, and thereby obtain the new assignment of images for sections $j+1$ through m.

However, a consequence of performing rank aggregation for each section independently is that the same image may appear more than once in a chapter. Consider a chapter consisting of two sections and suppose that we want two images for every section. Assume that the optimal assignments (ranked lists) corresponding to the two image mining algorithms are as follows. Alg_1(OPT): $s_1 \leftarrow \langle i_1, i_2 \rangle, s_2 \leftarrow \langle i_3, i_4 \rangle$ (that is, image i_1 has the highest relevance score and i_2 has the second highest score for section s_1 and similarly $\langle i_3, i_4 \rangle$ in that order are the top two images for section s_2), and Alg_2(OPT): $s_1 \leftarrow \langle i_3, i_4 \rangle, s_2 \leftarrow \langle i_1, i_2 \rangle$. Then the rank aggregation would give: $s_1 \leftarrow \langle i_1, i_3 \rangle$, $s_2 \leftarrow \langle i_1, i_3 \rangle$.

Algorithm 8 (Ensemble) avoids this problem by taking advantage of the logical linear organization of sections within a chapter. It considers sections in a chapter sequentially from the first section to the last, ensembling at a section level, and then removing images selected for this section from the pool of available images for the remaining sections. Before moving to a subsequent section, it reruns the image assignment optimization for the remaining sections over the remaining images. Thus images discarded due to merging for a section are taken into account for consideration in subsequent sections as such images may be more relevant than any of the candidate images for a section.

Consider a chapter consisting of three sections and suppose that we want two images for every section. Assume that the images and their relevance scores for different sections found by the two image mining algorithms are as follows. Alg_1: $s_1 \leftarrow \langle (i_1, 1), (i_2, 0.9) \rangle$, $s_2 \leftarrow \langle (i_7, 0.7), (i_8, 0.6) \rangle$, $s_3 \leftarrow$

$\langle (i_2, 0.5), (i_3, 0.4), (i_5, 0.3) \rangle$, and Alg_2: $s_1 \leftarrow \langle (i_3, 1), (i_4, 0.9) \rangle$, $s_2 \leftarrow \langle (i_7, 0.6), (i_8, 0.4) \rangle$, $s_3 \leftarrow \langle (i_4, 0.5), (i_1, 0.4), (i_6, 0.3) \rangle$. The optimal assignments would be: Alg_1(OPT): $s_1 \leftarrow \{i_1, i_2\}$, $s_2 \leftarrow \{i_7, i_8\}$, $s_3 \leftarrow \{i_3, i_5\}$, and Alg_2(OPT): $s_1 \leftarrow \{i_3, i_4\}$, $s_2 \leftarrow \{i_7, i_8\}$, $s_3 \leftarrow \{i_1, i_6\}$. The rank aggregation for the first section would give: $s_1 \leftarrow \{i_1, i_3\}$, thereby dropping i_2 from Alg_1 and i_4 from Alg_2 respectively. We note that i_2 is more relevant than current assignments for section s_3 under Alg_1 and similarly, i_4 is more relevant than current assignments for section s_3 under Alg_2. The benefit of rerunning the optimization is that such dropped images can be assigned to later sections (s_3 in our example). ENSEMBLE would result in the final assignment: $s_1 \leftarrow \{i_1, i_3\}, s_2 \leftarrow \{i_7, i_8\}, s_3 \leftarrow \{i_2, i_4\}$, which is more desirable than an assignment that excludes assigned images from later sections but does not rerun optimization ($s_3 \leftarrow \{i_5, i_6\}$).

6. ILLUSTRATIVE RESULTS

We now present some illustrative results from the empirical evaluation of the proposed techniques [4; 5; 6]. The corpus used in the study consisted of high school textbooks published by the Indian National Council of Educational Research and Training. It included books from grades IX–XII, covering four broad subject areas, namely, Sciences, Social Sciences, Commerce, and Mathematics. We selected this corpus as these book are used by millions of students every year and are freely available online.

6.1 Diagnosis of Deficient Sections

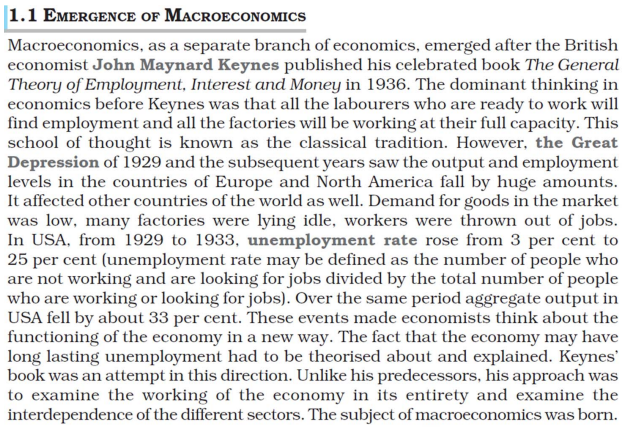
When applied to the corpus under study, our techniques were able to identify those sections of the books that could benefit from revision. The sections with high predicted scores for the need for revision often had a combination of large dispersion values (close to unity), lengthy sentences (up to six standard deviations to the right of the mean), and a large number of complex words.

One such section was found in Grade XII Sociology book and titled "Variety of Methods". We can see from the concept graph shown in Fig. 1(c) that this section has a number of disparate concepts leading to large dispersion. In addition, the section contains many long sentences, making the comprehension hard, *e.g.*: "Interviews may be structured, that is, follow a pre-determined pattern of questions or unstructured, where only a set of topics is pre-decided, and the actual questions emerge as part of a conversation."

Sections not needing update typically had low dispersion values (up to eight standard deviations left of the mean). The concept graph for one such section from Grade XI Mathematics book is given in Fig. 1(a). They also had simpler sentence structure making it easier for the reader to grasp the material well.

6.2 Augmentation with Articles

Fig. 2 reproduces the section titled "Emergence of Macroeconomics" from Grade XII Economics books. Our approach identified 'Macroeconomics','unemployment rate', 'Keynes', 'economics', 'Great Depression', 'Goods', and 'Demand' as the key concepts occurring in the section and and proposed the link to the Wikipedia article titled "Great Depression" as the best link for augmenting the section, though there is a Wikipedia article titled "Macroeconomics". If one reads the section carefully, one would notice that the large part



1.1 Emergence of Macroeconomics

Macroeconomics, as a separate branch of economics, emerged after the British economist **John Maynard Keynes** published his celebrated book *The General Theory of Employment, Interest and Money* in 1936. The dominant thinking in economics before Keynes was that all the labourers who are ready to work will find employment and all the factories will be working at their full capacity. This school of thought is known as the classical tradition. However, **the Great Depression** of 1929 and the subsequent years saw the output and employment levels in the countries of Europe and North America fall by huge amounts. It affected other countries of the world as well. Demand for goods in the market was low, many factories were lying idle, workers were thrown out of jobs. In USA, from 1929 to 1933, **unemployment rate** rose from 3 per cent to 25 per cent (unemployment rate may be defined as the number of people who are not working and are looking for jobs divided by the total number of people who are working or looking for jobs). Over the same period aggregate output in USA fell by about 33 per cent. These events made economists think about the functioning of the economy in a new way. The fact that the economy may have long lasting unemployment had to be theorised about and explained. Keynes' book was an attempt in this direction. Unlike his predecessors, his approach was to examine the working of the economy in its entirety and examine the interdependence of the different sectors. The subject of macroeconomics was born.

Figure 2: A section from Grade XII Economics

of the section (starting from "However, the Great Depression..." to "...33 per cent") describes the calamitous effect of great depression. Interestingly, we found after careful examination that the book had no other section where there was even a mention of great depression. We also examined the corresponding Wikipedia article and found that it contained information that a curious student would find very valuable.

We also identified the following two images for augmenting this section: (a) the famous Dorothea Lange's 1936 painting of migrant mother that depicts destitute pea pickers in California during great depression, and (b) image of John Maynard Keynes. Clearly, these images can further enhance understanding of this section.

6.3 Augmentation with Images

Fig. 3 shows the top five images proposed for three different sections from three different subjects. We can see that the images are quite relevant. We discuss the first example in more depth. This example shows the proposed augmentations for the section on how organisms create exact copies of themselves, appearing in the eighth chapter titled "How do organisms reproduce" in the grade X Science book. This section discusses three main points: (1) due to evolution, organisms are similar in their blueprint; (2) DNA replicates to pass on genetic material; and (3) DNA copying during reproduction should be consistent so that the organism is well adjusted to its ecosystem. We observe that the proposed images convey related information. The image on Phylogenetic trees captures the evolutionary relationships among biological species. The two images of DNA (chemical and physical structure) are illustrative of how the DNA can be easily replicated by breaking its double Helix structure. The section describes the consistency requirement of DNA copying using bacteria as the example organism. The images of RecBCD pathway in E. coli bacterium are complementary as it plays crucial role of initiating recombinational repair of potentially lethal double strand breaks in DNA.

We also conducted a user study employing the Amazon Mechanical Turk platform [28]. Seven judges each, coming from a population of 56 judges, judged the results produced by our implementation for a random sample of 100 textbook sections. The results demonstrate the promise of the pro-

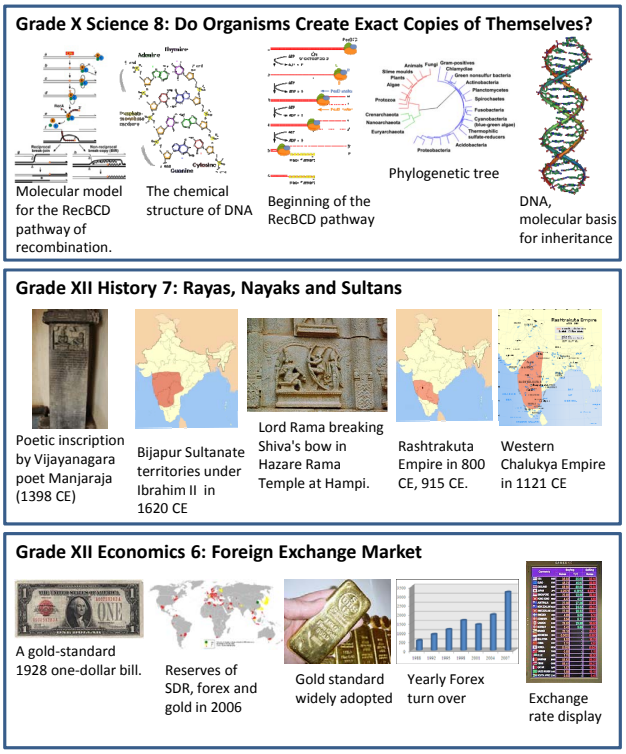


Figure 3: Augmentation with Images

posed system: the judges conservatively considered 87% of the images assigned to various sections to be helpful for understanding the corresponding section and the performance was maintained across subjects.

7. CONCLUSIONS AND FUTURE DIRECTIONS

Given the centrality of education for economic growth and the role of textbooks in a high quality education system, we set out to devise technologies for enhancing textbooks. We presented a diagnostic tool for algorithmically identifying those sections of a book that are not well-written and hence should be candidates for revision. We also described techniques for algorithmically augmenting different sections of a book with links to selective articles and images mined from the Web. We carried out an empirical evaluation of the proposed techniques using a corpus of high school textbooks published by the Indian National Council of Educational Research and Training. The preliminary results are promising and indicate that significant benefits can accrue by bringing to bear data mining technologies for improving the quality of textbooks. However, much more remains to be done.

A crucial element of our approach is the ability to identify the key concepts present in a section of the textbook. Drawing upon the NLP literature, we defined concepts to be terminological noun phrases [31]. However, it is worthwhile investigating other definitions, *e.g.* using ideas from the Formal Concept Analysis [20]. Similarly, additional techniques such as discourse analysis [34] can be applied for locating candidate concepts in the text.

Education researchers concur that the good textbooks are organized in a systematically progressive fashion so that students acquire new knowledge and learn new concepts based on known items of information [32; 43]. Many textbooks, however, suffer from the "mentioning" problem that causes concepts to be encountered before they have been adequately explained [9]. The diagnostic tool we presented operates at section level, treating each section independently, and does not address the flow of writing across different sections. A tool for diagnosing the comprehension burden due to non-sequential presentation of concepts would be valuable. More generally, designing tools for quantifying the quality of books combining multiple dimensions such as hierarchical organization, sequentiality, coherence and readability and then providing actionable recommendations for improvement is a fruitful direction.

Another promising direction is to examine what new issues arise if the ideas from this paper were to be extended for embellishing textbook material with other media types, *e.g.* video. There is also the related issue of selecting most appropriate type of augmentation across media types and tailoring the augmentations to suit the knowledge and experience level of the reader.

An obstacle we faced in our work was the lack of an established evaluation methodology for studying the performance of the proposed techniques. Ensuring objectivity and consistency across judges in the user studies are some challenges to be addressed in designing a direct measurement. Discounting externality and removing bias are some challenges to be addressed in designing an indirect measurement such as comparison of performance scores of students using good and bad versions of the same book.

We presented techniques for proposing articles and images with which a section of a textbook can be augmented, but did not discuss specific mechanisms for integrating the augmentations into the textbook. Our techniques could be integrated into authoring tools for helping textbook authors decide what materials to use when writing or revising a book. They can also be used for creating supplementary material that is distributed with the paper version of the books. Furthermore, there are ongoing efforts aimed at creating platforms and inexpensive devices for distributing books in a digital form (see, for example, the use of interactive DVDs as an educational platform [19], inexpensive e-book readers [3], and mobile learning devices [29]). Our work fits quite naturally with these efforts, but details need to be worked out.

Complementary approaches and related issues that merit serious future investigation include: (a) refining and enhancing the results produced by our techniques using collaboration and crowdsourcing [1; 44], (b) implications for royalty sharing and intellectual property rights [15], and (c) integration with other interventions for improving the learning outcomes [22; 37].

8. REFERENCES

[1] *Improving India's Education System through Information Technology*. IBM, 2005.

[2] S. Abney. Parsing by chunks. *Principle-based parsing*, 1991.

[3] A. Adams and J. van der Gaag. First step to literacy:

Getting books in the hands of children. *The Brookings Institution*, January 2011.

[4] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *CIKM*, 2011.

[5] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Identifying enrichment candidates in textbooks. In *WWW*, 2011.

[6] R. Agrawal, S. Gollapudi, K. Kenthapadi, N. Srivastava, and R. Velu. Enriching textbooks through data mining. In *ACM DEV*, 2010.

[7] J. Anderson and J. Pérez-Carballo. The nature of indexing: how humans and machines analyze messages and texts for retrieval. Part II: Machine indexing, and the allocation of human versus machine effort. *Information Processing & Management*, 37(2), 2001.

[8] K. Bakewell. Research in indexing: more needed? *Indexer*, 18(3), 1993.

[9] M. Chambliss and R. Calfee. *Textbooks for Learning: Nurturing Children's Minds*. Wiley-Blackwell, 1998.

[10] M. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, 2002.

[11] J. Coiro, M. Knobel, C. Lankshear, and D. Leu, editors. *Handbook of research on new literacies*. Lawrence Erlbaum, 2008.

[12] E. Coke and E. Rothkopf. Note on a simple algorithm for a computer-produced reading ease score. *Journal of Applied Psychology*, 54(3), 1970.

[13] A. Csomai and R. Mihalcea. Linking educational materials to encyclopedic knowledge. In *AIED*, 2007.

[14] E. Dale and J. Chall. A formula for predicting readability. *Educational research bulletin*, 27(1), 1948.

[15] L. Downes. *The laws of disruption: Harnessing the new forces that govern life and business in the digital age*. Basic Books, 2009.

[16] W. DuBay. *The principles of readability*. Impact Information, 2004.

[17] I. Fang. By computer: Flesch's reading ease score and a syllable counter. *Behavioral Science*, 13(3), 1968.

[18] C. Fellbaum. *WordNet: An electronic lexical database*. MIT Press, 1998.

[19] K. Gaikwad, G. Paruthi, and W. Thies. Interactive DVDs as a platform for education. In *ICTD*, 2010.

[20] B. Ganter, G. Stumme, and R. Wille. *Formal concept analysis: Foundations and applications*. Springer, 2005.

[21] J. Gillies and J. Quijada. Opportunity to learn: A high impact strategy for improving educational outcomes in developing countries. *USAID Educational Quality Improvement Program (EQUIP2)*, 2008.

[22] P. Glewwe, M. Kremer, and S. Moulin. Many children left behind? Textbooks and test scores in Kenya. *American Economic Journal: Applied Economics*, 1(1), 2009.

[23] S. Gollapudi and R. Panigrahy. Exploiting asymmetry in hierarchical topic extraction. In *CIKM*, 2006.

[24] W. Gray and B. Leary. *What makes a book readable*. University of Chicago Press, 1935.

[25] E. A. Hanushek and L. Woessmann. The role of education quality for economic growth. *Policy Research Department Working Paper 4122, World Bank*, 2007.

[26] M. Hu, E. Lim, A. Sun, H. Lauw, and B. Vuong. Measuring article quality in Wikipedia: models and evaluation. In *CIKM*, 2007.

[27] S. Huston and W. B. Croft. Evaluating verbose query processing techniques. In *SIGIR*, 2010.

[28] P. G. Ipeirotis. Analyzing the Amazon mechanical turk marketplace. *ACM Crossroads*, 17(2), 2010.

[29] A. Jawa, S. Datta, S. Nanda, V. Garg, V. Varma, S. Chande, and M. K. P. Venkata. SMEO: A platform for smart classrooms with enhanced information access and operations automation. In *International Conference on Next Generation Wired/Wireless Advanced Networking*, 2010.

[30] E. B. Johnsen. *Textbooks in the Kaleidoscope: A Critical Survey of Literature and Research on Educational Texts*. Scandinavian University Press, 1992.

[31] J. S. Justeson and S. M. Katz. Technical terminology: Some linguistic properties and an algorithm for indentification in text. *Natural Language Engineering*, 1(1), 1995.

[32] D. Kieras and C. Dechert. Rules for comprehensible technical prose: A survey of the psycholinguistic literature. Technical Report TR-85/ONR-21, University of Michigan, 1985.

[33] B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *KDD*, 1997.

[34] D. Marcu. Discourse trees are good indicators of importance in text. In I. Mani and M. Maybury, editors, *Advances in Automatic Text Summarization*. MIT Press, 1999.

[35] W. McCall and L. Crabbs. *Standard test lessons in reading*. Columbia University Teachers College Press, 1926.

[36] P. Menon. Mis-oriented textbooks. *Frontline*, August 2002.

[37] J. Moulton. How do teachers use textbooks and other print materials: A review of the literature. *The Improving Educational Quality Project*, 1994.

[38] N. Mulvany. *Indexing books*. University of Chicago Press, 2005.

[39] S. Panjwani, L. Micallef, K. Fenech, and K. Toyama. Effects of integrating digital visual materials with textbook scans in the classroom. *International Journal of Education and Development using Information and Communication Technology*, 5(3), 2009.

[40] C. Papadimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Dover, 1998.

[41] D. Saari. *Decisions and elections: Explaining the unexpected*. Cambridge University Press, 2001.

[42] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[43] R. Seguin. The elaboration of school textbooks. Technical report, ED-90/WS-24, UNESCO, 1989.

[44] B. W. Speck, T. R. Johnson, C. P. Dice, and L. B. Heaton. *Collaborative writing: An annotated bibliography*. Greenwood Press, 1999.

[45] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL-HLT*, 2003.

[46] A. Verspoor and K. B. Wu. Textbooks and educational development. Technical report, World Bank, 1990.

[47] K. Wang, C. Thrasher, E. Viegas, X. Li, and P. Hsu. An overview of Microsoft Web N-gram corpus and applications. In *NAACL-HLT*, 2010.

[48] A. Woodward, D. L. Elliott, and C. Nagel. *Textbooks in School and Society: An Annotated Bibliography and Guide to Research*. Garland, 1988.

[49] World-Bank. *Knowledge for Development: World Development Report: 1998/99*. Oxford University Press, 1999.

[50] S. E. Wright and G. Budin. *Handbook of Terminology Management*. John Benjamins, 2001.

[51] X. Xue, S. Huston, and W. B. Croft. Improving verbose queries using subset distribution. In *CIKM*, 2010.