Semi-Supervised Techniques for Mining Learning Outcomes and Prerequisites

Igor Labutov
Carnegie Mellon University
Machine Learning Department
ilabutov@cs.cmu.edu

Yun Huang
University of Pittsburgh
Intelligent Systems Program
yuh43@pitt.edu

Peter Brusilovsky
University of Pittsburgh
School of Information Sciences
peterb@pitt.edu

Daqing He
University of Pittsburgh
School of Information Sciences
dah44@pitt.edu

ABSTRACT

Educational content of today no longer *only* resides in textbooks and classrooms; more and more learning material is found in a free, accessible form on the Internet. Our long-standing vision is to transform this web of educational content into an adaptive, web-scale "textbook", that can guide its readers to most relevant "pages" according to their learning goal and current knowledge. In this paper, we address one core, long-standing problem towards this goal: identifying *outcome* and *prerequisite* concepts within a piece of educational content (e.g., a tutorial). Specifically, we propose a novel approach that leverages *textbooks* as a source of distant supervision, but learns a model that can generalize to arbitrary documents (such as those on the web). As such, our model can take advantage of any existing textbook, without requiring expert annotation. At the task of predicting outcome and prerequisite concepts, we demonstrate improvements over a number of baselines on six textbooks, especially in the regime of little to no ground-truth labels available[1]. Finally, we demonstrate the utility of a model learned using our approach at the task of identifying prerequisite documents for adaptive content recommendation — an important step towards our vision of the "web as a textbook".

1 INTRODUCTION

Amazon.com sells nearly 3 million textbooks, subsuming knowledge of virtually every discipline known to man. The information contained in these books, however, goes beyond the explicit knowledge recorded by the authors. Embedded in these books, is less tangible, implicit knowledge of how instructors *communicate* new concepts through previously explained concepts.

The value of this knowledge is amplified as more up-to-date educational material finds itself outside textbooks, and on the web [3]. The knowledge of how experts communicate new concepts through writing, reflected in the textbooks they write, provides the valuable data source for training machine learning models that can be deployed on educational content *outside* textbooks (e.g., we can identify which concepts the author of a web tutorial explains and which they assume as prior knowledge). **In this paper, our focus is on exploiting textbooks as a rich source of supervision for training such models.**

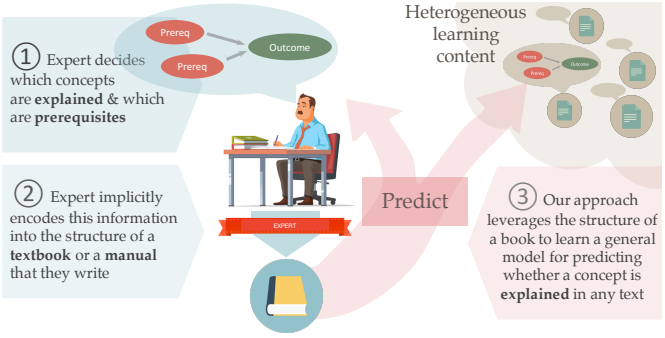


Figure 1: Textbooks are a natural source of supervision for learning a classifier to predict *outcomes* and *prerequisites* in educational texts. Our work explores the rich knowledge encoded in the structure of books to learn such classifiers.

The problem with textbooks as a source of supervision, however, is that they are inherently noisy. Rarely do authors explicitly state which concepts they explain in each section (i.e., *outcome* concepts) and which they assume as *prerequisites* – the kind of annotation that could be directly exploited to learn a classifier for discriminating between *outcome* and *prerequisite* concepts. Even if they did, the annotation would likely be inconsistent across different textbooks and authors. Instead, authors tend to reveal this information in an implicit way, via indirect signals that manifest themselves in the structure of the textbook. Two such indirect signals that we exploit in this work are:

- **Supervision Source 1: *Unit Cohesiveness***
  Our hypothesis is that the author usually explains a concept in one place (e.g., a chapter or a section).
- **Supervision Source 2: *Unit Titles***
  Our hypothesis is that the author of a textbook is more likely to include the concept's name in the title of a unit (e.g., chapter or section) if the concept is an *outcome* concept.

Neither of these signals, of course, will directly reveal whether a concept is an *outcome* or a *prerequisite*. A concept may sometimes be explained (i.e., an *outcome*) even if it's not mentioned in the title (Source 2). Similarly, knowing that a concept is explained

[1] textbooks are used for both evaluation and training

KDD '17, August 13-17, 2017, Halifax, NS, Canada

DOI: 10.1145/3097983.3098187

in one place says nothing about where (Source 1). Nevertheless, our hope is that these signals provide a clue, even if a noisy or incomplete one, that can be exploited by a model that is intimately aware of how these indirect signals and the concept labels (i.e., *outcome* vs. *prerequisite*) are ultimately connected. For our task of learning a *prerequisite*/*outcome* classifier (that can be deployed on any document), this entails building a tractable statistical model that (i) can exploit these indirect signals as sources of distant supervision and (ii) can also take advantage of any potentially available labeled data.

Historically, adaptive educational hypermedia systems relied on the notion of *outcomes* and *prerequisites* in implementing adaptive navigation support techniques [1] that aid users in navigating educational content. Almost exclusively, however, the annotation of concepts as *outcome* or *prerequisite* has been performed by experts – a major obstacle to automating and scaling adaptive hypermedia systems to the web. We believe that by developing a technique for learning *prerequisite*/*outcome* classifiers from unlabeled textbooks, our work offers an important step forward in the field of Adaptive Tutoring. Because our method relies on natural sources of supervision (in contrast to expert annotation of *outcomes* and *prerequisites*), available in any textbook, our method can take advantage of the millions of available textbooks with little to no additional effort. The main contribution of our work is the framework for leveraging unlabeled textbooks as a source of supervision, more specifically:

- A model that takes advantage of the titles of textbook units as weak labels, in order to learn a *prerequisite*/*outcome* classifier.
- A model that uses the intuition that a concept is usually explained in one unit as a (soft) constraint during learning, in order to learn a *prerequisite*/*outcome* classifier.
- A comprehensive evaluation of the two models at the task of *prerequisite*/*outcome* classification on a corpus of six textbooks.
- An end-to-end evaluation of the two models at the task of predicting *prerequisite* documents within a web textbook annotated with a prerequisite graph.

2 RELATED WORK

Active research on concept analysis for electronic textbooks started with the emergence of adaptive textbooks in late 1990 [7]. Adaptive textbooks attempted to use adaptive navigation support techniques from the field of adaptive hypermedia [1] to guide the reader to the textbook sections that are most appropriate for his of her level of knowledge and learning goals. To model reader's knowledge, adaptive textbook used traditional overlay knowledge modeling approach [4, 15] developed in the field of Intelligent Tutoring Systems. With this approach, domain knowledge is modelled as set of knowledge components frequently called *concepts* and the current level of user knowledge is independently measured for each of these concepts. A learning goal could be also represented in terms of this domain model as a subset of concepts to master. In this context, a section in an electronic textbook could be recommended if we know, which concepts this section is *explaining*. Indeed, if some concepts explained in a section are a part of the current learning goal, but not yet mastered, the section become desirable and can be recommended to the user. To support this kind of personalization, adaptive textbooks allowed their creators to indicate the explained concepts (known as *outcome concepts*) for each section of the book.

However, outcome concepts alone are not sufficient for good personalization. While a page might present some desirable outcome concepts, it might not be ready for the reader to learn since some earlier concepts (usually referred as *prerequisite concepts*) have to be learned first. To address these problems, a number of early textbooks such as ISIS-Tutor [5], KBS-Hyperbook [13], and Multibook [19] used network domain models with prerequisite links directly connecting domain concepts. Unfortunately, this approach originally developed for knowledge sequencing in small-domain Intelligent Tutoring Systems has not been scaling well for adaptive textbooks. Developing network domain models was notoriously hard. Besides, the nature of textbook explanations frequently required authors to refer to prerequisite concepts that were not anticipated by prerequisite links in the domain model.

To address this problem, some adaptive textbooks such as ELM-ART [6] and 2L670 [12] suggested to indicate prerequisite concepts directly for every textbook section. The model of an adaptive textbook where every section is indexed with a set of prerequisite and outcome concepts became very popular and was used by a number of platforms for authoring Web-based adaptive textbooks such as InterBook [2], AHA! [11], and NetCoach [20]. These platforms have been used to develop adaptive textbooks for a number of topics. The prerequisite/outcome model has been also used in some *open corpus* adaptive hypermedia (OCAH) systems [3]. The goal of an open corpus approach is to use in an adaptive way any online resource not originally considered by authors of an adaptive system. Indexing an online resource with a set of prerequisite and outcome concepts is one of the easiest ways to achieve this goal [9, 14].

The modern trend in OCAH system is gradual transition from manual content indexing to automatic concept extraction that allows a remarkable increase of scalability [18]. At the moment, many good concept-extraction approaches have been suggested. However, automatic separation of prerequisite and outcome concepts is still a problem with very few explored solutions. Past research explored two ideas to achieve this goal: the use of author's knowledge encapsulated in a sequential organization of a course [8] and more recently use of machine learning techniques to predict the class of each concept (*prerequisite* or *outcome*) on the basis of contextual and local features [10, 16, 17]. All of these methods, however, rely on datasets manually annotated with *prerequisites* and *outcomes*. Our key contribution in this work is a method for learning such classifiers from unlabeled textbooks, allowing us to potentially leverage millions of textbooks available in any discipline.

3 METHOD

In this section, we propose two probabilistic graphical models that we claim capture the intuition behind the two sources of distant supervision outlined in the previous section. A first step towards that, however, is to introduce the underlying classification model for predicting *prerequisites*/*outcomes*, which will be subjected to these two modes of distant supervision.

3.1 *Prerequisite/Outcome* classifier

The fundamental assumption underlying our work is that the authors of instructional content (e.g., a textbook) use different speech acts when referring to concepts that are *outcomes* vs. those that are *prerequisites*. Intuitively, we expect an explanation of a concept to involve acts such as defining or giving examples, which in turn are communicated via act-specific vocabulary (e.g., *such as*, *is a*, *is an example of*, etc.). More generally, we expect that the linguistic *context* surrounding the mentions of a particular concept is a good indicator to differentiate between whether the author is explaining that concept (i.e., *outcome*) or assuming it as *prerequisite* knowledge.

Naturally, this suggests a way of exploiting the linguistic context of a concept to automatically classify whether that concept is an *outcome* or a *prerequisite*. Using logistic regression as a model for classification, the probability that concept i in unit j (e.g., a chapter, section, etc.) is an *outcome* concept ($y_{ij} = outcome$) can be expressed as follows:

$$P(y_{ij} = outcome \mid \mathbf{x}_{ij}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_{ij})} \tag{1}$$

where $\mathbf{x}_{ij}$ represents the features of the context of concept i in unit j (e.g., chapter or section) and $y_{ij} \in \{outcome, prerequisite\}$. The features encoded by $\mathbf{x}_{ij}$ may include things like neighboring words (unigrams, bigrams, etc.), grammatical roles (e.g., subject, object), and anything else that might be relevant in helping classify whether a concept is an *outcome* or a *prerequisite*. However, because $\mathbf{x}_{ij}$ summarizes the context of a concept across the entire unit, but a concept is usually mentioned multiple times and using multiple terms (e.g., *EM* and *Expectation Maximization*), it is important to be precise about how the contexts of the individual mentions of a concept are aggregated to build a single unit-level representation of that concept.

A simple and straightforward way to aggregate the context features across the mentions of that concept within a unit is a linear combination:

$$\mathbf{x}_{ij} = \sum_{\substack{\text{mention } k \\ \text{of concept } i}} \mathbf{x}_{ij}^{(k)} \tag{2}$$

where $\mathbf{x}_{ij}^{(k)}$ is a feature representation of the context of the k^{th} mention of concept i in unit j. In all of our experiments in this work, we consider a single sentence to represent the context of a mention. Note that we may choose to transform the resulting linear combination to arrive at the final set of features for the concept (e.g., we may wish to binarize the features, as we do in our experiments).

3.2 Learning w via distant supervision

In this work, our core task is to learn $\mathbf{w}$, the parameters of the logistic regression classifier. In principle, having learned $\mathbf{w}$, we hope to be able to deploy this classifier on educational content beyond the textbook on which it was trained — an important step towards understanding the educational content on the web.

To this end, we now turn to the problem of learning $\mathbf{w}$. Recall that our core contribution are two methods that do not rely on the explicit annotation of *prerequisites* and *outcome* labels (i.e., y_{ij}) and instead employ weaker and more natural forms of supervision based on the universal structure of textbooks. Before describing the two models in technical detail, we briefly summarize how these models translate our high-level intuition about textbooks as a source of supervision, into the formal language of statistical modeling.

- SeqModel: ***Unit Sequence* as a constraint**
  Hypothesis: Our hypothesis is that the author typically explains a concept in one location (unit) within a textbook (i.e., units are cohesive in their focus).
  Model: This can be interpreted as an output-space constraint on the labels y_{ij} across all units j for a specific concept i. In the next section, we show that this constraint can be re-formulated as a tractable conditional mixture model.
- TitleModel: ***Unit Titles* as weak labels**
  Hypothesis: Our hypothesis is that the author of a textbook is more likely to include the concept's name in the title of a unit if that concept is an *outcome* concept, therefore making titles a source of weak supervision.
  Model: We model titles as noisy labels, i.e., corrupted version of the true labels y_{ij} which we treat as latent variables.

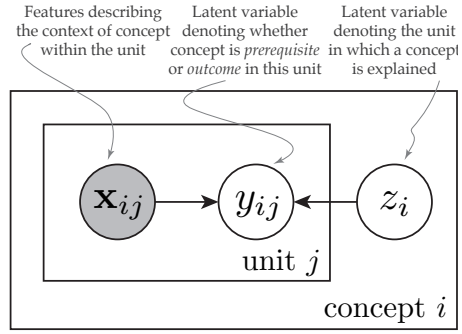
A note on outcomes and prerequisites. Note that by employing a binary classifier to distinguish between *outcomes* and *prerequisites*, we are making a strong assumption about a strict dichotomy, i.e., if a concept is not an *outcome*, it's necessarily a *prerequisite* and vice versa. In practice, of course, this is sometimes not the case. A concept may be referenced in passing, for example, without being either. At the same time, a concept in a single unit may appear as both, an *outcome* and a *prerequisite*. The assumption of a strict dichotomy is a convenient modeling assumption which often also holds in practice. In Section 5.5, we provide further justification to this assumption.

3.3 Supervision source 1: *Unit sequence*

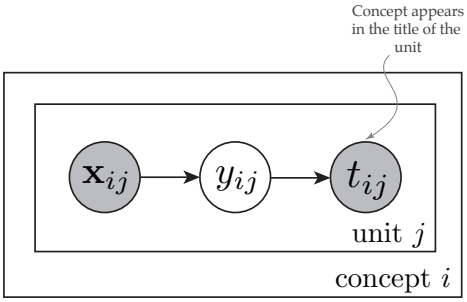
We begin by recalling one of our core assumptions about textbooks: *authors typically explain a concept in one location (unit) within a textbook*. Translating this assumption to the language of our model in Equation 1, we can interpret it as a constraint over a sequence of *prerequisite*/*outcome* labels y_{ij} across all units for every concept. To make this more explicit, consider the likelihood of observing all *prerequisite*/*outcome* labels $\mathbf{y}$ in a single textbook (i.e., for all concepts and units):

$$P(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = \prod_{\text{concept } i} \prod_{\text{unit } j} P(y_{ij} \mid \mathbf{x}_{ij}, \mathbf{w}) \tag{3}$$

Estimating $\mathbf{w}$ is straight-forward when the *prerequisite*/*outcome* labels $\mathbf{y}$ are available — precisely *not* the case for a vast majority of textbooks. Therefore, instead, we are going to rely on a weaker form of supervision, in the form of a constraint on the output space (i.e., labels $\mathbf{y}$), encoding our intuition that a writer typically explains a concept in one unit. We can express the optimization problem for



(a) SeqModel: ***Unit Sequence***



(b) TitleModel: ***Unit titles***

Figure 2: Graphical model instantiations of the distant supervision signals naturally present in textbooks.

estimating $\mathbf{w}$, taking this assumption into consideration, as follows:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{maximize}}\ P(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) \\ &\text{subject to}\ \sum_j y_{ij} = 1,\ \forall i \\ &\qquad y_{ij} \in \{0, 1\},\ \forall i, j \end{aligned} \tag{4}$$

Directly introducing an arbitrary constraint into the learning process is not trivial. However, we observe that a constraint of this form can be reformulated as an unconstrained conditional mixture model by introducing a latent variable. Consider a mixture model of N components, where N is the total number of units in the book. A component of this mixture can be expressed as follows:

$$P(\mathbf{y}_i \mid z_i, \mathbf{x}_i, \mathbf{w}) = P(y_{i,z_i} = outcome \mid \cdot) \prod_{j \neq z_i} P(y_{ij} = \neg outcome \mid \cdot)$$

where the latent variable $z_i \in \{1, \ldots, N\}$ indicates the unit where concept i is an *outcome* concept. Note that the above expression only defines a probability of observing a specific sequence $\mathbf{y}_i$ in which the *outcome* unit is given by z_i. Defining a full likelihood over all possible sequences $\mathbf{y}_i$ (given z_i) would entail specifying a probability over all permutations of assignments to the individual y_{ij} units in the sequence (i.e., for observations where the concept is explained in a unit other than the unit indicated by z_i). However, we avoid having to define the likelihood of these other observations by postulating that each concept is observed as an *outcome* precisely in the unit given by z_i. Thus, although the sequence $\mathbf{y}_i$ is not observed directly, we can nevertheless condition on it, in computing the posterior over z_i (which will be used in the E-step described below).

The likelihood over all concepts and units in Equation 3 can then be expressed as follows, taking into account the latent variables:

$$P(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = \prod_{\text{concept } i} \sum_{\text{unit } j} P(\mathbf{y}_i \mid z_i = j, \mathbf{x}, \mathbf{w}) P(z_i = j)$$

which can be interpreted as a classic admixture model, on which we can bring to bear the standard EM inference procedure. A representation of this model as a probabilistic graphical model is given in Figure 2a. Note that by performing EM inference, the constraint in Equation 4 is implemented in a "soft" way, i.e., the result of inference is a distribution over units where a concept is explained. We briefly describe the key steps of the algorithm:

E-step: The E-step computes a posterior distribution over the latent variables z_i (recall that z_i is an indicator variable, indicating the unit in the book where a concept i is explained). The expectation of the joint log-likelihood is then taken with respect to this posterior. After some algebra, we can show that the posterior over z_i is:

$$P(z_i = j \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{w}) = \frac{P(y_{ij} = outcome \mid \mathbf{x}_{ij}, \mathbf{w})}{\sum_{j'} P(y_{ij'} = outcome \mid \mathbf{x}_{ij'}, \mathbf{w})} \tag{5}$$

The posterior has a natural interpretation: the likelihood that a concept i is explained (i.e., *outcome*) in unit j is simply the normalized likelihood that the concept is an outcome concept in unit j given by the logistic regression model in Equation 1.

M-step: The M-step is a straight-forward weighted logistic regression problem, where the weights correspond to the posterior over z_i. We refrain from additional technical detail due to space constraints.

3.4 Supervision source 2: *Unit titles*

We now shift our attention to the second core assumption about textbooks: *authors are more likely to include the concept in a title if that concept is explained (i.e., outcome).* Translating this assumption to the language of our model in Equation 1, we can interpret titles as "weak labels" that are noisy, but correlated with the true labels y_{ij}. Let $t_{ij} \in \{0, 1\}$ be a binary variable that indicates whether or not the concept i in unit j appears in the title (e.g., *Introduction to Expectation Maximization*). The convenience of using t_{ij} is that they are always observed, while the true labels y_{ij}, indicating whether the concept is a *prerequisite* or an *outcome*, are not (i.e., latent). Figure 2b formalizes this intuition in a probabilistic graphical model.

Now, in addition to the parameters $\mathbf{w}$ of the logistic regression model, we need to introduce two additional parameters that connect the latent y_{ij} to the observed t_{ij}. These parameters are $P(t_{ij} \mid outcome)$ and $P(t_{ij} \mid prereq)$, defined in Table 1 below. Our intuition suggests that the author is more likely to include the concept in the title if they intend to explain or introduce the concept, rather than assumes it as a prerequisite. However, we expect that the model discovers by itself, the degree to which the appearance of a title correlates with whether the concept is an *outcome* or a *prerequisite* (via the two parameters $P(t_{ij} = 1 \mid outcome)$ and $P(t_{ij} = 1 \mid prereq)$). By decoupling the titles and the *prerequisite*/*outcome* labels within our model, we hope that the titles "nudge" the learning towards the correct model, while preventing them from contaminating it with noise.

$P(t_{ij} = 1 \mid outcome)$	Probability that the concept's name appears in the title if that concept is an *outcome*
$P(t_{ij} = 1 \mid prereq)$	Probability that the concept's name appears in the title if that concept is a *prerequisite*

Table 1: The two parameters connecting the appearance of a concept in the title and its *prerequisite*/*outcome* label.

We abstain from describing the E-step, as it is fairly straightforward to derive for this model. For learning the logistic regression parameters $\mathbf{w}$, the M-step also reduces to the weighted logistic regression problem, similar to the model in the previous section. For the $P(t_{ij} = 1 \mid outcome)$ and $P(t_{ij} = 1 \mid prerequisite)$ parameters, interestingly, the M-step can be performed in closed form, while also providing some insight into the model. An updated estimate of $P(t_{ij} = 1 \mid outcome)$ is computed as follows:

$$\frac{\sum_{i,j} \delta[t_{ij} = 1] P(y_{ij} = out \mid \cdot)}{\sum_{i,j} \delta[t_{ij} = 1] P(y_{ij} = out \mid \cdot) + \sum_{i,j} \delta[t_{ij} = 0] P(y_{ij} = out \mid \cdot)}$$

where $\delta[t_{ij} = 1]$ evaluates to 1 if concept i appears in the title of unit j, and 0 otherwise (we omit the $\mathbf{x}_{ij}$ and $\mathbf{w}$ on the conditioning side of $P(y_{ij} = out \mid \mathbf{w}, \mathbf{x}_{ij})$ with $\cdot$ to make the notation more compact). During the M-step, the above expression updates the estimate of $P(t_{ij} = 1 \mid outcome)$ by computing the fraction of times that the concept's name appeared in the title while the model also believed that the concept was an outcome (weighted by the confidence of the model). The same argument applies to estimating $P(t_{ij} = 1 \mid prereq)$.

During inference, we may choose to incorporate our prior belief about what $P(t_{ij} = 1 \mid outcome)$ and $P(t_{ij} = 1 \mid prereq)$ should be, which may help bias learning towards the right solution in the regime of little to no training data (less than 10% of data). For example, we may wish to encode our belief that $P(t_{ij} = 1 \mid prereq)$ should be very small (i.e., probability that the concept appears in the title when it is a prerequisite). Our experiments experiments show that even with weak priors, proposed model can work reasonably well. A natural choice for incorporating this prior knowledge is via a Beta distribution, which is a distribution on probabilities. In Section 5, we will describe our choice for the Beta prior parameters and their impact on inference in more detail.

4 EVALUATION

A natural metric to evaluate a *prerequisite*/*outcome* classifier (and the model used to learn it) is a performance measure such as AUC on held-out data with labeled *outcomes* and *prerequisites*. To this end, we annotate six textbooks with *prerequisite* and *outcome* concepts in each unit, and use these textbooks for both training and evaluation. In Sections 5.1 and 5.2, we discuss an evaluation of the two proposed models (SeqModel and TitleModel) at the task of predicting *outcome* and *prerequisite* concepts.

Recall, however, that our ultimate goal is to train a *prerequisite*/*outcome* classifier that can be deployed to educational text outside of the textbooks that were used to train it. To this end, in Sections 5.4 and 5.5, we explore the ability of the learned models to generalize to other textbooks and tasks. In the following section, we briefly describe the datasets that we used in our evaluations.

4.1 Datasets

In total, we employ seven textbooks for our evaluation: *Chris Bishop's Pattern Recognition and Machine Learning* (from hereon referred to as *PRML*), five OpenStax[2] textbooks in *Biology*, *Anatomy*, *Chemistry*, *Psychology* and *Economics*, and an online statistics textbook from Rice University[3], from hereon referred to as *StatsBook*. The *PRML* and *OpenStax* textbooks are used for evaluating the two models (SeqModel and TitleModel) at the task of predicting *outcomes* and *prerequisites*. The *StatsBook* is used to evaluate the utility of the predicted *prerequisite* and *outcome* concepts at a downstream task of identifying prerequisite units (documents) in a prerequisite graph. We briefly describe each corpus, emphasizing the process by which we obtained the ground-truth annotation in each textbook.

4.1.1 Bishop's PRML. We employ the index of the textbook to identify concepts, and the units where they are explained. *PRML* conveniently provides explicit expert-annotation of the units (section or subsection) where concepts are explained, by highlighting the page corresponding to that unit in the index. To identify concept mentions in the text of the textbook, we perform manual terminology normalization, i.e., we compile a list of terms that correspond to the same concept (e.g., *hidden variables*, *latent variables*). The advantage of manual term normalization is that we are able to leverage substantially more contextual features around mentions that would otherwise be missed via exact term matching. While we perform normalization manually for this dataset, we expect that this task would be automated in the future.

4.1.2 OpenStax textbooks. *OpenStax* textbooks provide a similar type of expert-annotation as *PRML*. Each unit in an *OpenStax* textbook contains a list of keywords, corresponding to the concepts that are explained in that unit. We employ these keyword lists at the beginning of each unit as expert labels, labeling the concept corresponding to the keyword as an *outcome* concept. We do not, however, perform any manual terminology normalization, as we did in *PRML*. We will discuss the consequences of this in Section 5.3.

4.1.3 Rice Statistics book. The Rice University online Statistics textbook (*StatsBook*) is unique in that it comes with an expert-created *prerequisite* graph. Every unit in the book is annotated with a set of other units that the author considers to be prerequisites (i.e., we can interpret this as a directed graph, where an edge exists between two units if one is a prerequisite of another). This type of annotation provides an opportunity to evaluate our model at a different, but related task, of identifying prerequisite units (documents). This task is closer to our broader goal of building on the results of the *prerequisite* and *outcome* classification, to create an automatic tutoring system that can guide learners through content.

Table 2 shows the statistics of these datasets.

[2] https://openstax.org/
[3] http://onlinestatbook.com/

Textbook	Units	Terms	Instances	Out	$P(t_{ij} \mid out)$
Bishop PRML	261	254	3,883	222	30%
Biology	47	637	3,188	637	1.9%
Anatomy	28	1,063	4,850	1,063	1.1%
Chemistry	22	234	1,358	234	3.0%
Psychology	16	185	801	185	5.9%
Economics	35	217	1,449	217	5.5%
StatsBook	113	340	1,879	—	—

Table 2: Textbook dataset statistics

5 EXPERIMENTS

In all experiments, we control the amount of ground-truth labels revealed to the model during training. This simulates the scenario where the annotator provides a small set of seed labels, and the model relies on both: (i) either the distributional constraint (SeqModel) or the weak labels (TitleModel) and (ii) the revealed ground-truth labels, in order to learn a *prerequisite*/*outcome* classifier. The focus of our evaluation, however, will be on the regime of very few labeled examples, which we consider to be more realistic in a practical setting. In all our evaluations, we perform 10 fold cross-validation, and report average AUC at the task of predicting *outcome*/*prerequisite* labels.

5.1 Supervision source 1: *Unit sequence*

We employ the following models in evaluating the SeqModel:

- **Semi-supervised**: SeqModel with EM-based learning, employing a variable amount of ground-truth labels during training.
- **Supervised**: A fully supervised logistic regression model, employing a variable amount ground-truth labels during training (and ignoring unlabeled data).
- **Baseline**: A naive baseline that considers the unit of the first occurrence of a concept in the book to be the unit where the concept is an *outcome*. This baseline stems from a strong intuition that an instructor would often explain the concept the first time they mention it. A logistic regression model is then trained on the dataset constructed using this assumption.

5.1.1 PRML evaluation. Figure 4a illustrates the results for the three models. We observe that the **Semi-supervised** model outperforms the **Supervised** model dramatically in cases with almost no ground-truth labels. We also observe that the effect of additional training data is fairly small, indicating that the simple distributional constraint on the labels based on the structure of the textbook is a powerful enough constraint to induce the labels without explicit annotation.

We illustrate two interesting posterior distributions over z_i, which represents the location in the textbook where term i is explained. The illustrations are presented in Figure 3a and Figure 3b, and concern two concepts (*conditional independence* and *latent variable*, respectively). The x-axis in both figures corresponds to the linear ordering of the units within the textbook (left end-point corresponds to the beginning of the textbook), and blue stems reflect the probability that term i is explained in a given unit (from the posterior distribution over z_i). Red stems correspond to the units where the term is annotated as an *outcome* (i.e., ground truth). In Figure 3a, observe that the term is *not* explained in its first appearance in the textbook, where the model correspondingly assigns low probability. Although *conditional independence* is given a cursory introduction in the first unit (where the posterior assigns a significant portion of the probability mass), the model correctly assigns the greatest probability mass to the chapter on graphical models, where the concept of conditional independence is explained thoroughly.

5.1.2 OpenStax evaluation. We conduct the same evaluation on the five OpenStax textbooks, and present the results in Figure 5. We observe that the **Semi-supervised** model continues to excel over the **Supervised** model and the **Baseline**, specifically in the regime of few to no ground-truth labels.

5.2 Supervision source 2: *Unit titles*

In evaluating the TitleModel, we also compare the **Semi-supervised** and fully **Supervised** models. Additionally, we implement a naive **Baseline** that uses titles as a source of direct supervision, i.e.:

- **Baseline**: A naive baseline that considers the presence of a concept in the title to be a deterministic indicator that the concept is an *outcome* concept. A logistic regression model is trained on this data, employing a variable amount of ground-truth labels during training.

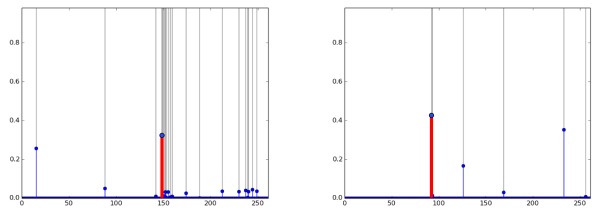
We find that with the exception of very few labeled examples (e.g., 4 examples), the model is insensitive to a range of different priors during learning.

5.2.1 PRML evaluation. Figure 4b illustrates the results for the three models. Similar to SeqModel, we observe that the **Semi-supervised** approach substantially outperforms the **Supervised** approach in the regime of little to no ground-truth data. Although the **Semi-supervised** approach outperforms the **Baseline** with a smaller margin when having little to no ground-truth data, it significantly outperforms the **Baseline** consistently across all regimes.

5.2.2 OpenStax evaluation. We conduct the same evaluation of the TitleModel on the five OpenStax textbooks. We find that while the **Semi-supervised** approach performs comparably to the best-performing baseline, it does not, however, show a significant performance gain for all except one textbook (*Economics*). In the next section, we carry out a detailed analysis via a simulation study, in order to explain the difference in performance between the *PRML* and *OpenStax* datasets.

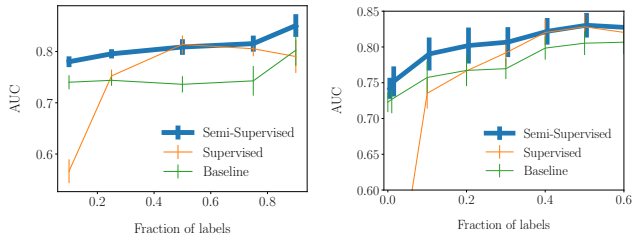
5.3 Why is the performance of TitleModel inconsistent between *PRML* and *OpenStax*?

From our experiments on the *PRML* and *OpenStax* datasets, we observe that the TitleModel has a noticeably different performance between the *PRML* and *OpenStax* datasets. From analyzing the statistics of the datasets in Table 2, our hypothesis is that the poorer performance of the TitleModel stems from a very low proportion of titles in which the concept's name appears when that concept is explained (i.e., low $P(title \mid outcome)$, last column in Table 2). For example, in *PRML*, approximately 30% of the units will mention



(a) Conditional independence (b) Latent variable

Figure 3: Posterior distributions over z_i, i.e., location (unit) in the textbook where a concept is explained (i.e., *outcome*). The x-axis corresponds to a linear ordering of the units in the *PRML* textbook. Red color identifies the ground-truth unit where the concept is claimed to be explained by the author of the book. See Section 5.1.1 for a discussion.



(a) SeqModel (b) TitleModel

Figure 4: Using the (a) unit sequence (SeqModel) and (b) unit titles (TitleModel) of the textbook as sources of distant supervision in learning a *prerequisite*/*outcome* classifier. Evaluation was performed on Bishop's Pattern Recognition and Machine Learning textbook (*PRML*). The performance of our model (*semi-supervised* in blue) provides significant gains over a fully supervised model (orange) especially in low training data regime.

the concept in the title if that concept is an *outcome*, in contrast to less than 6% in the *OpenStax* textbooks. To test our hypothesis, we conducted a simulation study, where we systematically generated synthetic "textbooks" with different $P(title \mid outcome)$ given a fixed $P(title \mid prereq)$, and compared the performance of the **Semi-supervised** model as a function of $P(title \mid outcome)$. Figure 6 summarizes the results.

We observe that the performance of the **Semi-supervised** model increases as a function of $P(title \mid outcome)$, substantially improving over the **Supervised** model in the regime of $P(title \mid outcome) > 10\%$, supporting our hypothesis that low $P(title \mid outcome)$ in *OpenStax* may be the reason for the **Semi-supervised** model's low gains over the baselines. Two natural questions arise: (i) why does $P(title \mid outcome)$ vary so substantially between the *OpenStax* and *PRML* textbooks and (ii) why does $P(title \mid outcome)$ affect the performance of the TitleModel? We provide our answers to these questions below.

Why does $P(title \mid outcome)$ ***vary so substantially between the OpenStax and PRML textbooks?*** Based on our analysis of the *PRML* and *OpenStax* data, we conclude that the lack of sufficient term normalization in the *OpenStax* data leads to low recall of concept mentions in the unit titles, and consequently an artificially lower $P(title \mid outcome)$ in the corpus. On the other hand, because *PRML* is processed manually (whereas terms are matched exactly in *OpenStax*), many more concept mentions are available in *PRML*, including mentions that appear in unit titles. To validate this claim, we have analyzed another textbook, Manning's Information Retrieval[4], where we manually normalized terminology, and annotated concepts as *prerequisites* and *outcomes*[5]. We have found that $P(title \mid outcome)$ is between 20% and 40%, depending on how the annotation of *prerequisites* and *outcomes* is performed. This further supports our hypothesis that term normalization is critical for obtaining a sufficient number of mentions of concepts in unit titles, and consequently a reasonable $P(title \mid outcome)$ required for the model to work well.

Why does $P(title \mid outcome)$ ***affect the performance of the*** TitleModel***?*** When $P(title \mid outcome)$ is low, there are simply too few titles to be exploited as weak labels during training ($P(outcome)$ also tends to be low ($< 25\%$) across all datasets). At the same time, because $P(title \mid prereq)$ is also consistently low across textbooks, both $P(title \mid prerequisite)$ and $P(title \mid outcome)$ become difficult to distinguish, making titles a less discriminative signal in learning.

5.4 How generalizable are the learned models?

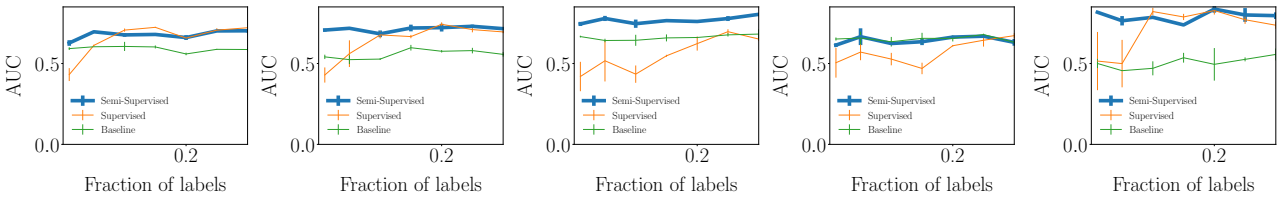
Recall that the core motivation for building a *prerequisite*/*outcome* classifier, is to be able to deploy it on educational content that exists outside textbooks, e.g., webpages. This raises an important question about the models proposed in this work: will a *prerequisite*/*outcome* classifier trained on one textbook, generalize to other content? To study this question, we carry out an evaluation where we train a *prerequisite*/*outcome* classifier on one textbook and evaluate it on the content of the remaining others. Although this task is still constrained to textbooks (because all of our annotated data consists of textbooks), the textbooks in our dataset are diverse, and present a reasonable challenge in testing the "generalizability" of the learned *prerequisite*/*outcome* models.

Figure 7 presents the results in a grid form, where rows and columns correspond to the textbooks that were used for training and testing respectively. We evaluate two versions of the SeqModel, trained with 10% and 90% of the ground-truth data, with AUC×100 displayed within the lower and upper parts of each cell respectively.

Based on these results, we draw the following two conclusions: (i) models learned using one source generalize to other content (even across diverse disciplines, such as *Anatomy* and *Economics*), and (ii) although models trained with more ground-truth data generalize better, they do so not by a large margin over models trained with only 10% of the data. Our key finding is that *the learned models are general, and capture some universal linguistic features of what it means to explain a concept, regardless of the concept's domain.*

[4] http://nlp.stanford.edu/IR-book/

[5] the textbook is not yet in a sufficiently processed state to be used as one of the datasets



(a) Anatomy (b) Biology (c) Economics (d) Chemistry (e) Psychology

Figure 5: Using the sequential structure of the textbook as distant supervision in learning a *prerequisite/outcome* classifier. Evaluation was performed on 5 OpenStax textbooks. The performance of our model (*semi-supervised* in blue) provides significant gains over a fully supervised model (orange) especially in low training data regime.

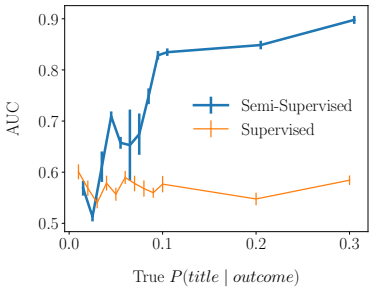


Figure 6: Simulation experiment varying the true $P(title \mid outcome)$ used to generate the synthetic "textbooks". We show that a sufficiently large $P(title \mid outcome)$ needs to be present in the dataset, for the model to be able to successfully exploit titles as weak labels during training.

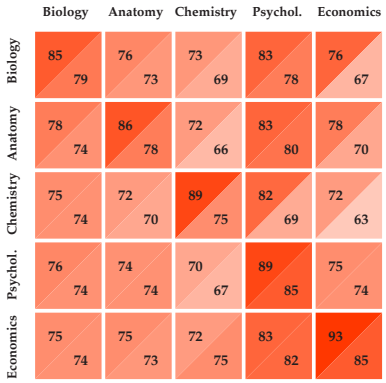


Figure 7: Evaluating the ability of the *prerequisite/outcome* classifiers trained on one textbook (row) to generalize to other textbooks (column). Upper and lower triangles correspond to classifier performance (AUC×100) trained with the SeqModel using 90% and 10% of the data respectively.

5.5 Can learned models predict prerequisite documents?

Recall that one of our goals is to use the *outcome* and *prerequisite* labels as input to a system that would guide the user through educational content. To this end, an important question is how the *prerequisite* and *outcome* labels can be used to discover prerequisite documents.

We follow the work of [17], who have created a dataset annotated with a ground-truth prerequisite graph, i.e., specifying how units in a textbook (*StatsBook*) are connected based on their *prerequisite/outcome* relations. A key connection between the task of predicting *prerequisite* documents (units) and *prerequisite* concepts (our work), is defining a function that computes a "prerequisite score" as a function of a pair of units, based on their *prerequisite* and *outcome* concepts (e.g., returned by our *prerequisite/outcome* classifier). Due to space constraints, we refer the reader to [17], who define a number of such scoring functions. In this section, we use the scoring functions and baselines that they develop in order to evaluate the output of the *prerequisite/outcome* classifier, trained using an approach proposed in this paper.

The task of predicting a *prerequisite* document (unit) in a textbook (*StatsBook*) can be posed as a binary classification problem, i.e., discriminating between documents that are *prerequisites* vs. those that are not (as done in [17]). Furthermore, we can analyze this classification performance as a function of the *prerequisite depth*, i.e., the distance between the original document and its prerequisite in the prerequisite graph, e.g., *Probability* is a prerequisite of *Expectation Maximization*, but is typically separated by many units in a textbook (large prerequisite depth) vs. *Maximum Likelihood* and *Expectation Maximization* which may follow each other closely (low prerequisite depth).

Figure 8 shows the performance of predicting *prerequisite* documents as a function of prerequisite depth for three different scoring functions:

- **Model**: this scoring function incorporates the output of the *prerequisite/outcome* classifier.
- **Baselines**: two naive baselines that compute a scoring function without using the outputs of the *prerequisite/outcome* classifier. See [17] for more detail.

Note that we train the *prerequisite/outcome* classifier on a different textbook (*PRML*), and deploy the learned model to this task. Note that prerequisite depth (x-axis in Figure 8) should be interpreted cumulatively, i.e., as all units at some prerequisite depth that is lower than x.

Based on Figure 8, we make the following observations: incorporating the *prerequisite/outcome* classifications from the TitleModel trained in a semi-supervised way improves performance (over the baselines) in predicting *prerequisite* documents. However, we find

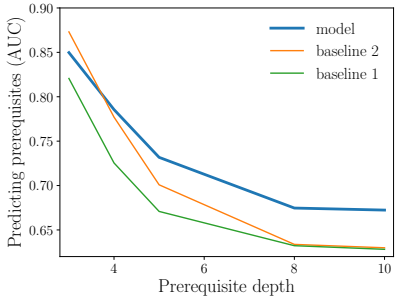


Figure 8: We evaluate the TitleModel at the task of predicting prerequisite documents in the *StatsBook* corpus annotated with a prerequisite graph. The TitleModel was trained on the *PRML* corpus with variable amount of ground-truth data. We find that fewer ground-truth labels (1% to 10%, with 10% shown in this figure) yields a model that is able to generalize better to this task. See Section 5.5 for details.

these performance gains exist only for a model trained with a small amount of ground-truth data (1% to 10% vs. 100%). We hypothesize that this is primarily due to the model trained with 100% of the ground-truth labels overfitting to the training textbook (*PRML*), and failing to generalize to *StatsBook*. Furthermore, in our experiments, *prerequisite*/*outcome* classifiers trained using the SeqModel did not substantially improve over the baselines in [17], which we too attribute to the model likely overfitting to the *PRML* dataset.

5.6 Discussion

A key take-away from our experiments and analysis is that the sequential structure of a textbook offers a more robust form of distant supervision than textbook titles. This stems from the fact that relying on the sequential nature of the book does not strongly depend on the quality and coverage of the extracted terms, as we have seen to be the case for a model that relies on titles as weak supervision. A model that relies on titles as weak supervision, will require a sufficiently high recall in term extraction, so that a sufficiently large proportion of title concepts are extracted. Any model that closely relies on terminology extraction will likely suffer from similar limitations.

More important towards our broader vision of deploying the learned *prerequisite*/*outcome* classifiers outside of the books on which they were trained, we demonstrate that the proposed methods yield classifiers that generalize to broader educational content.

6 CONCLUSION

Our key contribution in this paper is the idea and a practical demonstration of how textbooks can be leveraged as a rich source of supervision to train models for identifying learning *outcomes* and *prerequisites* in educational texts. We proposed two models that exploit two simple observations about how textbooks naturally encode experts' knowledge implicitly via the structure of the books they write, in order to learn the *prerequisite*/*outcome* classifiers without explicit annotation. This means that our method can utilize millions of published textbooks as training data.

Furthermore, we demonstrate that the *prerequisite*/*outcome* classifiers learned using our models generalize across diverse learning material (different disciplines), which implies that these models successfully capture some domain-agnostic essence of what it means to *explain* a concept. We believe that our work opens doors to a new generation of adaptive tutoring systems that make sense of the diverse educational content on the web (by understanding *outcome* and *prerequisite* concepts) and provide personalized guidance to learners to achieve their learning goals.

To encourage further research in this direction, we release all datasets annotated as part of this work, as well as all code that can be used to reproduce our results. They can be download at_this_url.

REFERENCES

[1] Peter Brusilovsky. 1996. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction* 6, 2-3 (1996), 87–129.
[2] Peter Brusilovsky, John Eklund, and Elmar Schwarz. 1998. Web-based education for all: A tool for developing adaptive courseware. In *Seventh International World Wide Web Conference*. 291–300.
[3] Peter Brusilovsky and Nicola Henze. 2007. Open corpus adaptive educational hypermedia. In *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer-Verlag, Berlin Heidelberg New York, 671–696.
[4] Peter Brusilovsky and Eva Millán. 2007. User models for adaptive hypermedia and adaptive educational systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer-Verlag, Berlin Heidelberg New York, 3–53.
[5] Peter Brusilovsky and Leonid Pesin. 1998. Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. *Journal of Computing and Information Technology* 6, 1 (1998), 27–38.
[6] Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber. 1996. ELM-ART: An intelligent tutoring system on World Wide Web. In *Third International Conference on Intelligent Tutoring Systems, ITS-96*. Springer-Verlag, 261–269.
[7] Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber. 1997. *Electronic textbooks on WWW: from static hypertext to interactivity and adaptivity*. Educational Technology Publications, Englewood Cliffs, New Jersey, 255–261.
[8] Peter Brusilovsky, Sergey Sosnovsky, Michael Yudelson, and Girish Chavan. 2005. Interactive Authoring Support for Adaptive Educational Systems. In *12th International Conference on Artificial Intelligence in Education, AIED'2005*. 96–103.
[9] Cristina Carmona, David Bueno, Eduardo Guzmn, and Ricardo Conejo. 2002. SIGUE: Making Web Courses Adaptive. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*. Springer-Verlag, 376–379.
[10] Sahar Changuel, Nicolas Labroche, and Bernadette Bouchon-Meunier. 2015. Resources Sequencing Using Automatic Prerequisite–Outcome Annotation. *ACM Transactions on Intelligent Systems and Technology* 6, 1 (2015), 6.
[11] Paul De Bra and Licia Calvi. 1998. AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia* 4 (1998), 115–139.
[12] Paul M. E. De Bra. 1996. Teaching Hypertext and Hypermedia through the Web. *Journal of Universal Computer Science* 2, 12 (1996), 797–804.
[13] Nicola Henze, Kabil Naceur, Wolfgang Nejdl, and Martin Wolpers. 1999. Adaptive hyperbooks for constructivist teaching. *Künstliche Intelligenz* 13, 4 (1999), 26–31.
[14] Nicola Henze and Wolfgang Nejdl. 2001. Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education* 12, 4 (2001), 325–350.
[15] Yun Huang, Michael Yudelson, Shuguang Han, Daqing He, and Peter Brusilovsky. 2016. A Framework for Dynamic Knowledge Modeling in Textbook-Based Learning. In *24th Conference on User Modeling Adaptation and Personalization*. ACM, 141–150.
[16] Sonal Jain and Jyoti Pareek. 2013. Automatic extraction of prerequisites and learning outcome from learning material. *International Journal of Metadata, Semantics and Ontologies* 8, 2 (2013), 145–154.
[17] Igor Labutov and Hod Lipson. 2016. Web as a textbook: Curating Targeted Learning Paths through the Heterogeneous Learning Resources on the Web. In *9th Intl. Conf. Educational Data Mining (EDM 2016)*. 110–118.
[18] Seamus Lawless, Lucy Hederman, and Vincent Wade. 2008. Enhancing Access to Open Corpus Educational Content: Learning in the Wild. In *The 19th ACM Conference on Hypertext Hypermedia*. 167–174.
[19] A. Steinacker, C. Seeberg, K. Rechenberger, S. Fischer, and R. Steinmetz. 1999. Dynamically generated tables of contents as guided tours in adaptive hypermedia systems. In *11th World Conference on Educational Multimedia and Hypermedia*. AACE, 640–645.
[20] Gerhard Weber, Hans-Christian Kuhl, and Stephan Weibelzahl. 2002. Developing adaptive internet based courses with the authoring system NetCoach. In *Hypermedia: Openness, Structural Awareness, and aptivity*. Springer-Verlag, Berlin, 226–238.