Peer-Based Intelligent Tutoring Systems: A Corpus-Oriented Approach

by

John Champaign

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis, we present an artificial intelligence approach for tutoring students in environments where there is a large repository of possible learning objects (e.g. texts, videos). In particular, we advocate that students learn on the basis of past experiences of peers. This aligns with McCalla's proposed ecological approach for intelligent tutoring, where a learning object's history-of-use is retained and leveraged to instruct future students. We offer three distinct models that serve to deliver the required intelligent tutoring: (i) a curriculum sequencing algorithm selecting which learning objects to present to students based on benefits to knowledge obtained by similar peers (ii) a framework for peers to provide commentary on the learning objects they've experienced (annotations) together with an algorithm for reasoning about which annotations to present to students that incorporates modeling trust in annotators (i.e. their reputation) and ratings provided by students (votes for and against) for the annotations they have been shown (iii) an opportunity for peers to guide the growth of the corpus by proposing divisions of current objects, together with an algorithm for reasoning about which of these new objects should be offered to students in order to enhance their learning. All three components are validated as beneficial in improving the learning of students. This is first of all achieved through a novel approach of simulated student learning, designed to enable the tracking of the experiences of a very large number of peers with an extensive repository of objects, through the effective modeling of knowledge gains. This is also coupled with a preliminary study with human participants that confirms the value of our framework. In all, we offer a rich and varied role for peers in guiding the learning of students in intelligent tutoring environments, made possible by careful modeling of the students who are being taught and of the potential benefits to learning that would be derived with the selection of appropriate tutorial content.

Acknowledgements

First and foremost I would like to thank my supervisor, Professor Robin Cohen, for guiding me through the completion of this dissertation and my PhD studies. Robin overwhelmed me with the amount of time and effort she put in to helping me complete my doctoral work. My committee members, Professor Edward Lank and Professor Peter van Beek, have provided detailed and helpful feedback throughout the process, which I greatly appreciate. Thanks as well to Professor Gord McCalla for his helpful advice. My appreciation goes out to the anonymous reviewers who considered earlier versions of this work and provided feedback during conference and journal submissions over the last 4 years.

I would like to acknowledge Ahmed Ataullah, Simina Brânzei, John Doucette and Lachlan Dufton for being wonderful officemates: always willing to talk through issues I was working on, go for coffee and share a laugh. I greatly enjoyed working in the AI Group at the University of Waterloo with a large number of intelligent, thoughtful and accomplished students and professors.

I owe my deepest gratitude to Amy Hamilton, Jason Kogan, Nicholas Miller, my family (Ralph, Wendy and Paul Champaign) and the Smith clan (Jim, Jettie, Chris, Janet and Danielle) for their support over the last 4 years. In addition to her support, I'd also like to thank Miranda Gabriel for encouraging me to pursue a PhD and for providing feedback on the psychological elements of our human study.

The Natural Sciences and Engineering Research Council of Canada (NSERC), NSERC's Strategic Networks of Research hSITE project, Ontario Graduate Scholarship (OGS) Program and the David R. Cheriton Graduate Scholarship, provided financial assistance for which I am grateful.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Providing an education has always been a part of human culture, evolving from more experienced members of a community helping less experienced members through various pedagogical strategies and classroom environments to the current state where worldwide governments invest the equivalent of 1.97 trillion U.S. dollars in education [78].

The field of intelligent tutoring systems (ITS) has existed since the 1960s, beginning with systems to have automated tutors emulate the personalized teaching offered by one-on-one instruction from real teachers (e.g. the SCHOLAR system of Collins and Carbonell[16]) and then progressing to efforts to model student learning of lessons, in order to diagnose and assist those students (e.g. Greer and McCalla, 1994[30]; Johnson and Soloway, 1984[39]), through to efforts to represent student models using the machinery of user modeling (Kay, 2008[42]). In order to direct the learning of a student using an intelligent, automated system, the content of lessons needs to be authored, typically with input from pedagogical experts. Most recently efforts have been developed aimed at allowing students to benefit from the learning that their peers are experiencing (e.g. Vassileva, 2008[84]). One central motivation for the design of peer-based intelligent tutoring systems is to cope with the cost of authoring an entire ITS. Research has convincingly shown (Murray, 1999[58]) that the primary limitation on widespread adoption of ITS is simply the cost of development.

This work attempts to address this cost of development, providing multiple techniques to assist in the development of intelligent tutoring systems. Our peer-based approach leverages past experiences of students using the ITS, rather than having instructional developers explicitly detail the content that each student should experience, based on a complex set of rules. Our aim is to deliver personalized education to students in a peer-based, ecological approach which focuses on (i) sequencing objects (e.g. text, video) from a repository (ii) enabling peer tutoring through the provision of annotations (additional commentary left by peers) and (iii) supporting peer-suggested division of learning objects, to allow the repository to grow. The models that we develop are examined and validated both in an abstract, simulated domain of instruction and in a concrete domain, with real students (home healthcare).

1.1 Our Approach in Detail

Beyond the cost of development, determining the best way to deliver educational content to students has been a rich domain of inquiry. Issues such as student modeling, interfaces for instruction, student affect, assessment, providing hints and assistance, and encouraging peer collaboration have each occupied extensive amounts of research effort.

In order for students to receive instruction, material to be presented to them must first be assembled. This involves an expensive process of working with educational and domain experts to elicit the information that needs to be taught and to fit it into the system in an appropriate way. It is desirable to personalize an experience, providing instruction in a manner that takes into account a particular student's strengths and weaknesses and teaches them in a way that best enables them to learn. Creating the infrastructure for this personalization becomes a critical challenge. Typically this involves modeling of the students, then having the system adapt itself to the various types of students using it.

McCalla has proposed an ecological approach for the design of intelligent tutoring systems [54] in which he advocates that student learning be achieved on the basis of the experiences of previous students but in a way that evolves over time, as the students themselves adjust with their learning.

Our approach builds on the idea of an ecological approach to peer-based instruction. Traditionally peer-based instruction has meant ways to reason about how to bring students together so that they can learn from one another. Works such as Read et al.'s [66] involved the creation of an ITS to form groups of students best able to help one another in learning a second language. Some systems, such as Lee's work on student discussions [48] have tried to leverage past interactions by showing to each new student dialogues of previous students with an instructor; this was intended to allow new students to learn by reading questions and answers from these past interactions.

Rather than simply finding and displaying past interactions between students and the system and rather than requiring real-time peer interaction, our approach uses data from a history of past interactions in order to reason about how best to provide a personalized experience to the current student. The central idea in this work is to allow the previous experiences of peers to form the basis for the selection of content for new students. Personalization arises by determining the similarity of the current student to previous students when selecting content, and giving a greater weight to previous students who are most similar. While showing content to students, our techniques also attempt to display those items which are expected to give the most benefit to the current student. How we determine and validate this expected benefit is the core of this work.

The repository (all the information on a topic that might be shown to a student) consists of a large number of possible objects[1]. By attaching a history of the experience with past students to these objects, it becomes possible to reason using active interpretation of this historical data. Once we are working with a repository, we can also integrate new avenues for student learning (provided by the student population that uses this corpus). We then need to validate that the student-produced content provides a better educational experience. We detail two approaches to this a) leaving annotations, which allows students to leave short text messages that may be useful for subsequent students using the learning object they are interacting with, and b) the corpus approach which allows students to suggest a division of a learning object, to produce a more streamlined version that removes unnecessary information and helps students focus on the core, vital information.

The techniques we propose work best with large amounts of data from a number of students taken over the course of their instruction. Conducting a large scale study with real students would be worthwhile, but incredibly challenging. In this work we detail an alternative validation approach, which we refer to as simulated students. By simulating learners, the material they are assigned, and the outcome of their interactions, we are able to perform a number of large scale studies which would otherwise be infeasible. Beyond the contributions from our techniques for engaging in intelligent tutoring, this approach to an early, inexpensive validation offers valuable insights for our research community. We provide details on how the simulations were performed; we also confirm the value of the simulated results by replicating the work with a preliminary human study.

It can be difficult to select the exact best curricular content for a student at the beginning of their educational experience. Inexperienced teachers quickly learn that there is a difference between knowing something and teaching it. It can be challenging for instructors to put themselves in the shoes of students who are encountering the material for the first time in order to provide a sequence of material to lead every student from ignorance to understanding. Once instructors have assembled material that provides such a course of instruction and provides answers to common questions, it then becomes challenging for

[1] We assume that some expert has assembled this repository and it is not a random collection.

every student to be expected to follow the same path through the material. Students who already understand part of the course of study are expected to quietly consume information they already understand, while students who are struggling and haven't mastered a concept are forced to move on to a higher level concept they don't have sufficient background to interpret.

Instead, by modeling the experiences of past students rather than requiring an explicit roadmap produced by a single, highly knowledgable instructor, we are able to direct the selection of curricular material. Peer feedback also allows the production of new content (in the form of annotations) and better targeted education content (in the form of new divided objects) to suggest a tailored sequence for students to consume the material. As will be seen in detail, we in fact not only show objects to students that they are expected to benefit from, but also we avoid showing objects to students that do not offer benefit or that may harm their educational progress. This is especially important when peers have a role to play in influencing the possible learning of a student. We therefore consider another central contribution of our work to be the promotion of a greater role for peers, carefully handled through effective reasoning about the content that they suggest.

Our research was partially funded by hSITE [63] (an NSERC Strategic Research Network in Healthcare Support through Information Technology Enhancements focused on delivering effective healthcare through the use of technology in environments that employ extensive networking and sensors). Because of this, we include an examination of the role of our techniques for peer-based intelligent tutoring in the home healthcare domain. To this end, our human study focused on education of primary caregivers for children on the autistic spectrum. Home healthcare has many challenges that would benefit from our work. Medical doctors are typically expected to be the primary source of health information for patients, however time constraints prevent them from doing much more than giving a pamphlet to their patients. Patients are understandably highly motivated to learn as much as possible about their diagnosis, but are often unable to obtain guidance for considering worthwhile, valid information. Sources such as the Internet provide material of varying quality. Our techniques allow a system for validating information and providing it in a tailored fashion while respecting the time limitations of professional healthcare workers. In this context, patients or their caregivers would be learning how to cope with a particular illness. To avoid self-directed care, wading through a massive set of opinions and documents offered online, intelligent tutoring offers promise while a peer-based approach still allows for the experience of those in similar scenarios to be included.

Intelligent tutoring systems have repeatedly been shown (see Section 8.5) to effectively assist in the education of students, but the utility of these systems has typically been demonstrated in modest, small-scale studies. The models presented in this thesis, validated for larger scale repositories through simulations, offer an important step forward.

In all, our work details multiple techniques which personalize an intelligent tutoring system using data from previous interactions with students, rather than requiring explicit, detailed directions from expensive, busy experts. Ultimately these techniques offer an approach that can be combined with any computer-based educational system to enhance and personalize the curriculum sequence. Beyond the ordering of material, the techniques provided allow a population of students to refine the corpus of material to better suit the particular needs of that learning community, and to make it possible for the tutoring to evolve over time such that the right information is provided to the right students and at the right time. Society places an ever increasing burden on its members to continually learn and adapt to the changing technology and culture. Education has become a necessity for many careers, with a large portion of these requiring life-long learning. With this in mind, our aim is to be offering to the next generation of students an automated learning environment that enables better education, while respecting the current attraction of connecting socially online with the experiences of peers.

1.2 Overview of Thesis

The remainder of the thesis includes the following chapters:

- (Chapter 2) Background: The necessary background information from other researchers in the community to put this work in context to provide a justification for some of our approaches to validation.
- (Chapter 3) Curriculum Sequencing: An overview of our approach to recommending educational content to students based on the past experiences of similar students
- (Chapter 4) Annotations: An overview of our approach to allowing students to leave short text messages for subsequent students and to explicitly rate the value of annotations that they have been shown, and then intelligently reasoning about the best annotations to show to a particular student, incorporating the inherent reputation of the annotator, ratings from other students, and the similarity between the current student and those past raters.
- (Chapter 5) Corpus Approach: An overview of our approach to allow student-directed subdivisions of content and the synergy between this technique and the curriculum sequencing to make recommendations about which student populations may benefit from the streamlined version.

- (Chapter 6) Driver: A short overview of how to combine the three core techniques (from the previous three chapters) into one comprehensive algorithm to deliver the overall benefit of an intelligent tutoring system.

- (Chapter 7) Additional Validation: A description of a human study which informed and provided validation for the curriculum sequencing, annotations and corpus approach.

- (Chapter 8) Discussion: A further exploration of the contributions this work has made, placing these results in the context of the ITS research community.

- (Chapter 9) Conclusion: A final summary of the work performed for this thesis and the presentation of an array of possible future directions.

Chapter 2

Background

2.1 Intelligent Tutoring System Overview

Our work is situated within the architecture of an intelligent tutoring system as described by Vassileva [81, 82, 85] is presented in Figure 2.1 and clarified in Table 2.1, 2.2 and 2.3[1].

The focus of our research is on the Planner, Teaching Materials, and Teaching Materials Editor components of Figure 2.1, utilizing data from the History. All the components of the intelligent tutoring system work together and produce a course which is presented to the student as output.

[1] Our description of the components of an ITS are our own interpretation and may not align exactly with Dr. Vassileva's view.

Role	Description
Author	The scientist in the upper left hand corner of the diagram in front of the blackboard represents the stakeholder who built the ITS.
Teacher	The man behind the desk in the upper right hand corner represents the stakeholder who is teaching the course and is responsible for the student's education.
Student	The boy at the bottom left cradling his head in his hands represents the stakeholder who is attempting to use the system to master the material in the course of instruction.

Table 2.1: Description of Stakeholders of an ITS Architecture

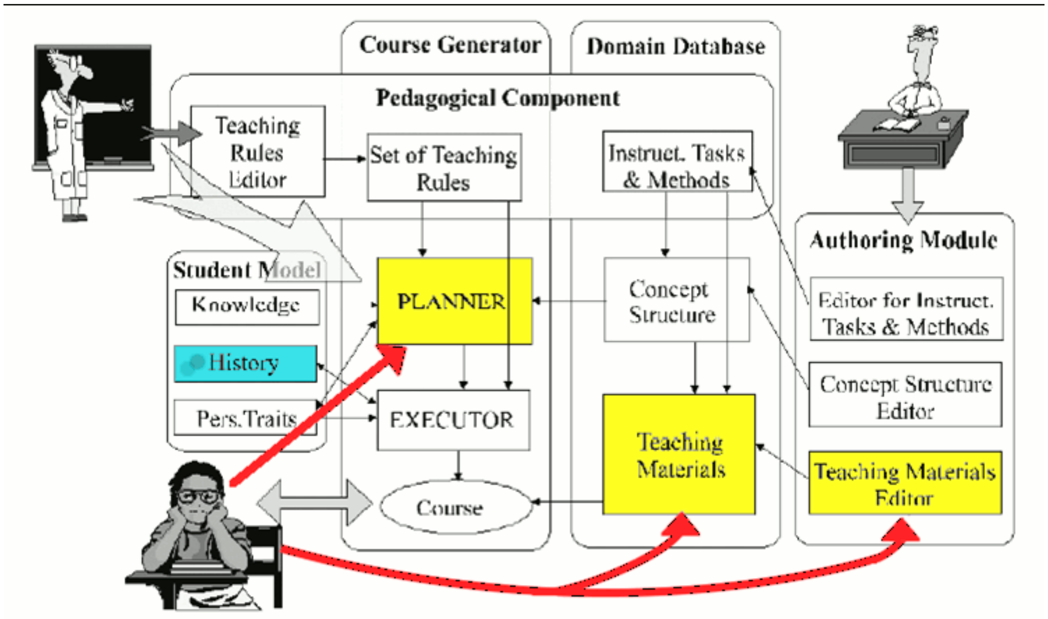


Figure 2.1: Intelligent Tutoring System Architecture [81]

Name	Description
Course Generator	Creates the course of study, controls interactions with the student and updates the Student Model
Domain Database	Representation of the subject of instruction and related learning materials
Pedagogical Component	A balance of generic instructional approaches and domain-specific knowledge that may have been customized by the teacher
Student Model	A representation of the student's preferences, understanding of the domain of instruction and a history of interactions
Authoring Module	Components which allow the instructor to customize the ITS for his particular course

Table 2.2: Description of Modules of an ITS Architecture

Name	Description
Teaching Rules Editor	A component that allows a teacher to hard-code a specific instructional strategy which will over-ride the Set of Teaching Rules and Instructional Tasks and Methods below
Set of Teaching Rules	The high-level, strategic approach the system follows while interacting with a student
Instructional Tasks and Methods	The specific, low-level, tactical plan for interactions with a student created by decomposing a high-level interaction strategy into concrete assignments of particular sub-tasks
Knowledge	A user model of how well the student understands the concepts in the course of study
History	A record of every interaction between the system and a particular student, including data that was used by the system to determine how to interact with the student
Personal Traits and Preferences	A model of the student's affect and various psychological elements
Planner	Creation of an actual course plan, including both what material to present to the student and in what order to present it
Executor	The implementation of the plan provided by the planner
Course	The combination of the Executor and the Teaching Rules Editor in order to carry out instruction of the student
Concept Structure	A model for domain of instruction and connections between concepts in that domain
Teaching Materials	The user interface for the system which contains the student-focused presentation, input and assessment portions of the ITS
Editor for Instructional Tasks and Methods	User interface for teacher which allows manual creation of pedagogical approaches, both high-level and low-level
Concept Structure Editor	Allows the creation of a model for domain of instruction and connections between the varying concepts within it
Teaching Materials Editor	The interface between the repository of learning objects and the implemented ITS (may be an API which learning objects need to conform to, for example)

Table 2.3: Description of Components of an ITS Architecture

2.2 The Role of Learning Objects in Intelligent Tutoring

In this section we clarify the concept of a learning object, which is central to our proposed intelligent tutoring approach. We begin with a description of research that helps instructors teach by providing them with a repository of useful objects. We then discuss research in the context of intelligent tutoring systems which suggests the value of marking up objects in a repository, in order for the system to intelligently reason about what to show students.

We take a broader definition of learning object compared to many other ITS researchers [75, 57, 84]. While we use the examples of text and video as learning objects throughout this thesis, we also embrace the idea of quizzes (with feedback on what the student got right or wrong), simulations, structured interactions between students learning together and even entire other intelligent tutoring systems as learning objects. Our idea of learning objects embraces anything that could be imagined being completed on a computer to help students learn. A course of study can be constructed by connecting a number of learning objects, which students complete in turn, with assessments in between the interactions.

2.2.1 Learning Objects As Guidance for (Non-Automated) Teaching

In South and Monson's work describing the deployment of a university wide learning object based system [75] the authors detail an extensive project to create a system for Brigham Young University in Provo, Utah[2]. The motivation for the creation of learning objects included:

- Massive labour and hardware costs associated with providing media for a wide number of media and formats
- Redundancy where more than one department would buy the same instructional materials (for example, two departments had 60% overlap in their slide libraries)
- Physical space limitations and the desire to move more instruction to distance and Internet-based course offerings

[2] To clarify, this is not an intelligent tutoring system, but instead uses learning objects in a system to help instructors author course content.

- To support hybrid courses, that met on a limited schedule and conduct the majority of course-work online
- Rising development costs and standards for instructional material
- Innovation which was restricted to the developer's course and therefore had limited impact

Their approach to the creation of learning objects was to create precise specifications, in terms of podium hardware and digital formats, in order to migrate to an all-digital system for delivery of course material. They used a single database as the sole repository for all instructional media. In order to maximize the reusability of learning objects, they found it was best for objects to be focused on a single, core concept. Larger learning objects at the course, lesson or module level tended to be harder to re-purpose. Conversely, without any context, tiny "learning objects" become unassociated media (such as an image or sound clip). They encouraged development of learning objects at the "sweet spot" of granularity.

As an example of learning objects, they have multiple learning objects focused on Newton's 1st Law of Motion which include: a slow motion car crash, an interactive simulation of a man "surfing" on ice in the back of a truck, and a series of questions about a woman in a moving elevator that assesses a learner's understanding.

For the actual authoring, they provided an instructional template where instructors could drag-and-drop self-contained media objects. They were, at the time of writing, planning to develop more sophisticated building templates that would provide a "wizard" for creation of learning objects.

They found a particular challenge in their approach was tagging learning objects with metadata, such as a tag that clarifies the medium (e.g. text or video) of the learning object or an identifier for the author. They involved library staff in this project, which gave them a greater consistency due to using professionally trained catalogers. However, even this was an overwhelming project and ultimately they had to minimize the number of fields required for each object and only provide metadata for collections of objects. They also encouraged creating metadata during the development of learning objects.

2.2.2 Metadata Approaches

One approach for assigning learning objects to students used in certain intelligent tutoring systems is to mark up the learning objects with metadata, perform student modeling, then use constraint satisfaction techniques to match a learning object with a student's need. Metadata is typically described as "data about data" while ontologies have been described as "an explicit specification of a conceptualization... the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them"[31]. For example, a health-care focused ontology[14] might include *terms* such as disease, illness, doctor and person, identify *synonyms* such as disease and illness, and detail *relationships* such as a "doctor is a person". Health-care focused metadata might be classifiers such as "Name", "Age", "Gender" for a person and "Symptoms", "Treatment", "Diagnosis" for a disease. In this work we take the simplifying perspective that ontologies are a specialized, more sophisticated, type of metadata.

Ontologies, as discussed in the work of Nkambou [95], are defined as:

> An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest, where formal implies that the ontology should be machine-readable and shared that it is accepted by a group or community. Further, it should be restricted to a given domain of interest and therefore model concepts and relations that are relevant to a particular task or application domain.[14]

This formal conceptualization can either be created manually [57, 75] or automatically [95], typically by natural language processing or machine-learning techniques, and is used to reason about matching learning objects to a curriculum sequence.

Morale and Aguera [57] detail their experiences building a prototype system that uses a metadata approach to delivering learning objects in a dynamic sequence. Their work involves a standard for learning object metadata which minimally involves: an identifier, a name, a location, a theme and a class. This metadata is used to reason about intelligently assigning learning objects to a student, which they do not clarify in full, but merely indicate as "The learning objects presupose[sic] the existence of an environment with the capacity to decide which object is to be presented next". The dynamic element in the title of their work refers to the capacity of their system to offer students the choice of a traditional or socratic sequencing. Traditional sequencing involves an organization defined by the course author and involves all explanations followed by all examples followed by all exercises followed by all tests followed by full feedback. The socratic sequence involves first testing students: if they show mastery of the test, another test is displayed. However if they don't show mastery, an exercise is instead displayed. The students are then re-tested and if they still do not show mastery they are then shown an example before being re-tested again. The authors assume that learning objects may be composed of other learning objects, a view we are sympathetic to and explore briefly in Section 9.2.4.3.

While enticing in theory, and achievable in focused domains within research labs, both the metadata and ontology approaches present numerous challenges.

Brooks and McCalla [10] provide extensive details, with examples, of the problems which may result when using learning object metadata in adaptive learning environments. Marking up learning objects with metadata is expensive and time consuming and there will be continual tension between the producers of learning objects, who desire a minimal well-defined set of metadata, and consumers who desire a rich, tailored selection of metadata. Much like software systems, an organization's changing needs or conflicting needs of numerous organizations result in a large demand for extensive data about each learning object. Inevitably, learning objects will be added to the system or modified, and the metadata becomes inconsistent and inaccurate. Many researchers still working with this approach attempt to automatically generate metadata, by reasoning about the learning object. Even if such reasoning is possible, avoiding the metadata and just reasoning directly about the learning object and students seems more straightforward.

2.2.3 McCalla's Ecological Approach

Since there are various challenges directing student learning on the basis of metadata attached to learning objects, it is useful to consider competing approaches to intelligent tutoring. McCalla's ecological approach to e-learning systems [54] is described as "attaching models of learners to the learning objects they interact with, and then mining these models for patterns that are useful for various purposes." Learning objects are used in the ecological sense, in that instead of attaching static ontologies to objects as metadata, models of previous interactions (i.e. presenting learning objects to students) are used in their place. These models are then actively interpreted, with the meaning derived by real-time processes, as information about the object is needed.

Part of the flexibility of the ecological approach is that data can be gathered before it is known to what purpose it will be put. This allows an existing system to have new capabilities added, which utilize the rich history of interactions with students.

In McCalla's ecological paradigm for intelligent tutoring system, the basis for tutoring is a repository of learning objects (which, for example, could be a chapter from a book, a video or an exercise which is completed and evaluated), which are provided to students for them to interact with in order to improve their learning. Information about these interactions is used to reason about the best learning objects to be shown in the future to other students, based on the utility of past interactions of similar students with the various objects in the system.

In order to design effective peer-based intelligent tutoring systems several challenges arise, however. Some of the central ones include:

- deciding which peers should be used to form the basis of the tutoring
- deciding which content should be presented to new students, based on their peers' previous learning

Since McCalla's ecological approach is intended to primarily be a general philosophy for designing intelligent systems, individual researchers need to then create their own algorithms and systems to embody this approach. This work is an effort to do so for the problem of deciding what content to show to students, including an allowable adjustment to the construction of repositories of learning objects by peers (following their learning experiences). The value of our approach is explored through a method of simulating student learning, followed by a preliminary study with human users.

2.3 Peer-Based Tutoring

Peer-based tutoring, where students are encouraged to interact and learn from one another, has been shown to be an effective learning strategy both in the classroom and in ITS. In the classroom, substantial learning can occur by encouraging students to discuss concepts among themselves. Helping their classmates understand the material lets more advanced students attain further mastery, while students struggling with the material get tailored, individual explanations.

ITS research has demonstrated the value of encouraging these collaborations when learning on a computer as well. Typically the approach has been to have students interact in a structured manner to maximize the benefit from the interactions. Several examples [32, 18, 84, 11, 66] are detailed in Section 8.5.1.2.

Our approach differs from this traditional perspective in that we attempt to use the experiences of past students to inform the interaction with a current student rather than encouraging interaction among multiple active students. We reason about similarities between students when doing so (the peer part of our approach), rather than just blindly using all past experiences with the system.

In addition to helping to inform student interactions with learning objects, our techniques allow students interacting with the system to influence the repository of learning materials and to modify what is available to be presented. This further supports the peer-based perspective in our work. [3]

Once our framework is presented in full, we provide a more detailed comparison with peer-based tutoring approaches in Section 8.5.1.2.

2.4 Recommender Systems

Recommender Systems [67, 33] use the opinions of a community of users to assist individuals in efficiently separating interesting content from a large variety of choices. The actual mechanics of a recommender system can vary widely such as: a binary "recommend or not", a recommendation on a scale ranging from love to hate, recommending sequences of items (such as a playlist of songs that will be particularly pleasing to a user), or finding credible recommenders (see Trust modelling in Section 8.5.5). Recommender systems can also operate implicitly or explicitly, anonymously, publicly, or with pseudonyms, with various models for aggregating content and reasoning about connections between individuals in the community and by combining recommendations with content analysis.

On the surface, it might seem that recommendation techniques could be applied directly in an ITS setting. However, whereas most recommender systems endeavor to obtain an increasingly specific understanding of a user, an ITS seeks both to understand a user and to change them. In addition, in contrast to positioning a user within a cluster of similar users, we would like to model a continually evolving community of peers where students at a lower level are removed and more advanced students are added as a student works through the curriculum.

2.4.1 Collaborative Filtering

Collaborative filtering [1, 7, 33] is a popular technique which uses advice of a group of users, for each new user. The most well-known commercial example of this is Amazon.com's feature which recommends products that were also purchased by customers who have purchased the item being considered. Collaborative filtering employs a set of techniques (such as correlation, vector-similarity, default rating, inverse user frequency, cluster models,

[3] We assume that the students interacting with the system are highly motivated to learn and participate in the learning community. In certain contexts, such as patient health education, this is a reasonable assumption. For other contexts we assume techniques from other work, such as COMTELLA (see Section 8.5.1.2) could be used to incentivize students.

or Bayesian network models) which use a database of user preferences to predict what a student might like [7]. Ultimately, what unifies the various approaches is that they use data about users' past experiences to predict the utility of items for a particular user: the active user.

Breese et al. [7] presents memory based collaborative filtering as a user database of ratings ($r_{i,j}$) representing user i's rating on item j where I_i is the set of items i has rated and the mean rating for user i is:

$$\overline{r_i} = \frac{1}{|I_i|} \sum_{j \in I_i} r_{i,j} \tag{2.1}$$

This allows a prediction of the expected rating the active user a will leave for item j ($p_{a,j}$) based on the weighted sum of other users' ratings:

$$p_{a,j} = \overline{r_a} + \kappa \sum_{i=1}^{n} w(a,i)(r_{i,j} - \overline{r_i}) \tag{2.2}$$

where n is the number of users who have nonzero weights, which reflect the distance, correlation or similarity between the active user and every other users in the database. Many alternatives exists for calculating this weight.

The intuition behind this formula is that we are starting with the average rating a user has left in the past, the $\overline{r_a}$, then modifying it based on the difference between what all other students have provided as a rating for the current item j ($v_{i,j}$) and the average rating each has left in the past ($\overline{r_i}$). This term, ($r_{i,j} - \overline{r_i}$), converts an absolute rating into a relative rating for that user. The similarity term, $w(a,i)$, lets us then give a weight to each students' rating based on the importance placed on user i's opinion, from the perspective of the active user a. κ is a normalizing factor included to ensure that the absolute value of the weights sum to 1.

In our research, we are exploring how the previous learning experiences of other students with an online repository of learning objects (texts, videos, articles, etc.) can be used to leverage the learning of current students in an intelligent tutoring environment. Inspired by the collaborative filtering approach of recommender systems [7], we aim to first of all identify like-minded students as the basis of the social network that is assisting the student. With this pool of students, we then aim to recommend to students the objects that have offered the most significant benefit to those previous peers. In contrast with other researchers, we are interested in considering previous learning experiences that may have already ended, not restricting ourselves to the networking of students who are currently engaged in learning, for our peer-based approach.

In particular we are focused on techniques and algorithms to produce recommendations in e-learning settings, addressing the challenge of how to provide suitable recommendations in the educational domain. After presenting our proposed approach in detail, we reflect further on the relationship between our model and more traditional recommender systems in Section 8.5.4, leading to some proposals for future research. Throughout, our aim is to elaborate on how to support peer-to-peer learning within educational recommender systems.

In our work, we used a memory-based collaborative filtering algorithm [7] using the Manhattan distance applied to a vector space of their assessments to reason about the similarity between students. The precise algorithm is detailed in Chapter 3.

An educational recommendation is more challenging than typical recommendations. In addition to gathering data to understand the student, the system needs to model how the student changes and make recommendations that are relevant to their current understanding and educational needs. We overcome the changing student issue by considering every assessment to be a unique individual when reasoning about what learning object to recommend to a student, so treating each student as if they were a new user each time a recommendation is made[4].

This decision has consequences, such as the fact that previous interactions between the same student and the system will influence recommendations. We do not model learning trajectories, such as how a student's overall learning progress is related to other students (our approach is a greedy, breadth first approach - reasoning about the best next step). The benefit of this approach is that it allows the system to be highly responsive and reason about new peers as soon as the student demonstrates competency.

2.5 Zhang's Approach to Evaluating Trustworthiness

Zhang and Cohen [94] propose an approach to evaluating recommendations provided by advisors in order to model the reputation of sellers in an e-commerce setting based on ratings provided by these past buyers (called advisors), allowing for the possibility for ratings to be unfair (that is, other agents in the marketplace may lie).

[4] We consider an extension to this assumption as future work in Section 9.2.2.5.

Their model assumes a marketplace populated by other buyers and sellers with varying degrees of "trustworthiness"[5]. Their work is designed around two dimensions of categorization: Public/Private and Local/Global. Private refers to the ratings provided by advisors who share ratings on particular sellers with the the active buyer such that the degree to which their ratings correspondence can be calculated[6]. "Public" refers to the degree to which a particular buyer's rating corresponds to the consensus rating (from all buyers in the system) for a particular seller. Zhang and Cohen also use Private and Public to contrast a buyer's personal experience with a seller and the reputation calculated by advisor ratings. Global refers to reputation systems which reason about ratings provided by buyers about all sellers in the system in order to estimate trustworthiness. In contrast, "Local" refers to a reputation system which only reasons about ratings that have been left on the seller being considered (rather than all sellers) when evaluating trustworthiness.

For the Public / Private portion of their work, which we incorporate into our annotation approach detailed in Chapter 4, they detail a personalized approach to evaluating the trustworthiness of other buyers (advisors) which uses the Chernoff Bound theorem[7] in order to determine the active buyer's confidence in an advisor. For their personalized approach, a private and public reputation is determined for all advisors, with the weight given to each reputation being determined by the number of mutual ratings evaluated using the Chernoff Bound theorem. For the extremes, advisors with whom the active buyer shares a large number of mutual ratings will be considered exclusively based on their private reputations (weight of 1) and their public reputation will be ignored (weight of 0). For advisors with whom the active buyer shares no ratings, their reputation will be based entirely on their public reputation (weight of 1) and the private reputation will be ignored (weight of 0).

When modeling the trustworthiness of sellers, Zhang and Cohen's approach balances the personal experiences of the active buyer with the recommendations of advisors. Similar

[5] In their work trustworthiness is used to refer to both sellers and other buyers (advisors) with slightly different connotations. A trustworthy seller can be counted on to provide truthful information about their product or service and to follow through on transactions. A trustworthy advisor can be counted on to provide truthful ratings based on their experiences with sellers.

[6] For example, if two buyers have both rated a particular seller as untrustworthy, they would assign one another a higher private reputation. If one buyer had rated the seller as trustworthy and the other had rated the seller as untrustworthy than the buyers would have a lower private reputation for one another.

[7] This statistical approach is used to determine the number of trials needed to differentiate between events with known probability: for example, for a biased coin that comes up with one side 60% of the time, determining which side it is biased towards. Contrast this with another biased coin where one side comes up 90% of the time. Chernoff bounds can be used to determine the number of trials needed to be confident in the categorization, perhaps 150 trials in the first case but only 10 trials in the second.

to the above process, a threshold number of experiences is used to balance the personal experiences with the reported reputation.

2.6 Overview of Our Proposed Corpus-Oriented Approach to Peer Tutoring

2.6.1 Curriculum Sequencing

In this thesis we use the term curriculum sequencing to refer to the following task: Given a set of learning objects and an associated history of interaction between previous students and those objects, which learning object should be assigned to the active student? Our proposed model for curriculum sequencing is the focus of Chapter 3.

We record with each learning object those students who experienced the object, together with their initial and final states of knowledge, and then use these interactions to reason about the most effective lessons to show future students based on their similarity to previous students. As a result, we are proposing a novel approach for peer-to-peer intelligent tutoring from repositories of learning objects.

Two primary elements were in focus:

i) specifying the algorithm that determines which peers and which learning objects are most important for each new student, based on a modeling of similarity matching and of the benefit derived from learning objects by previous peers

ii) presenting a validation of the approach that simulates student learning, leveraging an assessment in terms of letter grades (through pre- and post-tests) as well as a modeling of the target knowledge levels for each learning object.

We develop a preliminary solution for i) and then extend this to demonstrate the robustness of the algorithm by introducing error into the assessment used as part of the validation. Through this extension, we are able to show that the average knowledge level attained by the students continues to reflect appropriate learning, because the ongoing collaborative recommendation of learning objects helps to compensate for errors that are introduced.

We also explore a richer modeling of learning objects in terms of their expected time requirements. From here, we return to populate our simulations with learning objects of varying temporal demands, continuing to operate with possible errors in assessment as well. We also extend the size of the learning object repository to be much larger, in our experiments. We are able to show that our revised algorithm continues to provide high levels of knowledge to students, on average.

This work is relevant to the subtopics of personalized e-learning and adaptive educational systems, user contribution and peer-based assistance, group-oriented collaborative recommendation technologies and modelling the evolution of individual participation and social relationships. We return to discuss the contribution of our research to these subfields of study (Chapter 8).

2.6.2 Annotations

We use the term annotations to refer to a brief commentary attached to a learning object, left by a student who has experienced that learning object, of possible use to a subsequent student as part of their learning. Our proposal for annotations is the focus of Chapter 4. The central challenge we address is as follows: Given a learning object assigned to an active student, a set of annotations attached to that learning object and a history of ratings on annotations from all students in the system, which set of annotations should be shown to this particular student using this particular learning object?

To honour the basic ecological approach, we are particularly interested in exploring the use of student annotations: allowing students to leave short comments on learning objects they are interacting with. More than simply tags, this could be a question or a commentary about what they're learning. Subsequent students would identify which annotations they found useful, which would then be intelligently shown to similar students. Asynchronous collaboration or, at least, allowing the interactions of the student in the past to inform the interaction with the current student, honours the ecological approach [54].

Beyond implicitly reasoning about interactions (for the curriculum sequencing), we extended this paradigm to allow students to explicitly leave information for future students. This takes the form of "annotations", or short text messages, which are attached to the learning object and intelligently shown to (or hidden from) future students. As a motivating example, suppose a student Carol in a Computer Science 101 course was struggling with the concept of procedures. She suddenly realizes that procedure is another name for a function, which is the term her high school teacher had used. She leaves a short annotation ("OMG, I realized half way through the lesson that procedures are the same as functions!!! duh! :-)") on the learning object she was studying with this insight and carries on with her lessons. Future students are shown this annotation and given the option of endorsing it (by clicking on a "thumbs up") or denouncing it (with a "thumbs down"). Over time, the system learns that the annotation tends to be useful to students with a background that used the term "function" instead of "procedure" (and shows it to them), but not to others (and hides it from them).

2.6.3 Corpus Division

We use the term corpus division to refer to allowing students to propose that a shorter learning object, derived by extracting a division of a current learning object, should be added to the repository. Our proposal for corpus division is the focus of Chapter 5. The central challenge we address is as follows: Given a streamlined version of a learning object proposed by a student, how is it best to incorporate this into the repository and recommend it to future students?

Beyond curriculum sequencing and annotations, we further focus on the process of enabling peers to augment the corpus of learning objects, proposing subdivisions of existing objects as valuable for guiding the learning of subsequent students. This entails providing students with coarse, easy-to-use authoring tools which allow them to suggest a streamlined version of a learning object they have completed which may be more useful to students in the future. We provide an algorithm that reasons about which learning objects are best to offer each new student, with this new repository that includes both the original objects and the new, subdivided entries.

2.6.4 Simulated Student Learning

One of our two central techniques for validating our models is to simulate student learning. In order to do so, we require a model of the knowledge gains by students and a model of learning summarized below.

2.6.4.1 Explanation of Multi-Dimension Model of Knowledge

In this work we use a multi-dimensional model of knowledge to represent student understanding of a domain of study. Some subjects, such as mathematics or computer science build on themselves gradually, where a thorough understanding of one part of the field of study is necessary to understand a more advanced concept. For example, it would be very difficult to understand recursion without an understanding of the concept of a function. Similarly, it would be difficult to understand the addition of fractions without a strong understanding of the meaning and notation of fractions. In contrast, other fields such as history, literature or geography often cover disjoint topics that, while reinforcing a central core to a course, can each be understood independently from the others. One doesn't need a full understanding of the book "1984" in order to evaluate "Brave New World"; however, an understanding of one of these works complements a study of the other.

To model an abstract domain that can contain both these concepts of prerequisite and independent knowledges we used this multi-dimensional model. Each axis of understanding focuses on one part of the instructional domain (which can be thought of as a course at a school). Each axis ranges from 0, complete ignorance of the topic from the perspective of the course of study through 1, complete mastery of the topic of study from the perspective of the course.

Averaging all axes in the n-space (where n refers to the number of dimensions) gives a final value, ranging from 0 to 1, which is analogous to the final mark a student would receive if they were evaluated with their current understanding of the course material. This makes the simplifying assumption that each axis contributes equally to the final evaluation[8].

Using such a model allows us to consider courses of study where the material is tightly connected and builds on itself (which would result in a low number of dimensions), or courses of study with heterogeneous material that is loosely coupled (which would result in a high number of dimensions). Learning objects were modeled that would target varying understandings within a multi-dimensional model of knowledge and could (possibly) enhance understanding of one topic while harming understanding of another. This, again, is present in real world learning environments.

Example: The atomic theory is often first explained to students as electrons orbiting a nucleus, much as planets orbit a star (or moons orbit a planet). Given a general understanding of orbits, this gives physics students a model to think about parts of the atom by connecting it to other things they already understand, namely Newtonian motion. In reality, it has been known for decades through quantum mechanics that electrons do not orbit the nucleus like planets or moons, but in fact the "orbits" should instead be thought of as a probability distribution of where the electron might be at any moment. Even though the orbital model is incorrect, it is a simpler way to convey the atomic theory to students, and is therefore still presented. Using a pedagogical scaffold such as this is an example of something that harms the students' understanding of one concept, while helping another.

[8] This assumption could be relaxed, if it were desirable to model a field of study where various dimensions were given different weights. See Section 9.2.5.1 for further details.

2.6.4.2 Model of Learning

When running our algorithm in the simulation, each student would be presented with the learning object that was predicted to bring the greatest increase in learning, determined by selecting those learning objects that had resulted in the greatest benefit for previous students considered to be at a similar level of understanding as the current students.[9]

Simulated students allowed us to avoid the expense of implementing and experimenting with an ITS and human students to see the impact of our approach in contrast with alternative approaches. In particular we contrasted our method with a baseline of randomly assigning students to learning objects and to a "look ahead" greedy approach where the learning was pre-calculated and used to make the best possible match. One variant we considered was a "simulated annealing" inspired approach, where greater randomness was used during the initial, exploratory phase of the algorithm, then less randomness was used once more information about learning objects had been obtained. As will be shown, we discovered that our approach showed a clear improvement over competing approaches and approached the ideal.

2.6.5 Wizard of Oz Style Human Study

The second technique used to validate our models is an actual study with human users. For our human study (Chapter 7), we used what is known as a "Wizard of Oz" approach. From Human-Computer Interaction [44, 43, 3, 35]. The idea behind this approach is that participants interact with a system that is at least partially controlled by a human. The name is taken from the movie "The Wizard of Oz" where, at one point the heroine Dorothy and her companions encounter the all-powerful Wizard of Oz. It is discovered that, rather than being an all-powerful wizard, he is simply a man controlling a projection system using levers and a microphone.

In a typical "Wizard of Oz" study the participants are led to believe that the human controlled elements are actually provided by software. In our study we were upfront about what was being done by the computer and what was being done by the experimenter. Deception was unnecessary, as we were investigating the efficacy of the educational material presented rather than the participants' reaction to a fully-functional system. Instead of creating a complete computer-based educational environment, we wrote a program that would use the student assessments and our curriculum sequencing approach (see Section 3)

[9] Each student in the simulation is modeled to have a current level of understanding for each possible knowledge area, a value from [0,1] reflecting an overall grade from 0 to 100.

to recommend a learning object to the student. The recommended learning objects were then given (some as paper articles, others as videos played on a netbook) to the participants, while assessments were given as paper-and-pencil quizzes that were evaluated and entered into the system by the experimenter.

Chapter 3

Curriculum Sequencing

Given a number of learning objects, a history of interactions between the objects and past students and a student ready to be assigned work, which of the possible learning objects should be assigned to her? Our approach to this problem uses a collaborative filtering inspired algorithm to balance the similarity of the current student to past students with the benefit those past students have obtained, in order to create a personalized recommendation.

In this chapter we first introduce the approach and then present its validation using simulated student learning.

Our proposed algorithm for determining which learning objects to present to students is presented in Algorithm 1. We assume that we are tracking a set of values, *v[j,l]*, representing the benefit of the interaction for user j with learning object l. *v[j,l]* is determined by assessing the student before and after the interaction, and the difference in assessed knowledge is the benefit. We also record for each learning object the previous interactions of students with that object, in terms of their initial and final assessments. We assume that a student's knowledge is assessed by mapping it to 18 concrete levels: A+, A, A-, ... F+, F, F-, each representing $\frac{1}{18}$th of the range of knowledge. This large-grained assessment was used to represent the uncertainty inherent in assessing student knowledge, and only this large-grained assessment is used to reason about the students' ability in our approach.

The anticipated benefit of a specific learning object l, for the active user, a, under consideration would be: [1]

$$p[a,l] = \kappa \sum_{j=1}^{n} w(a,j)v(j,l) \tag{3.1}$$

[1] Adapted from Breese et al.'s collaborative filtering paper [7].

```
Input the current-student-assessment
for each learning object: do
  Initialize currentBenefit to zero
  Initialize sumOfBenefits to zero
  Input all previous interactions between students and this learning object
  for each previous interaction on learning object: do
    similarity = calculateSimilarity(current-student-assessment,
    interaction-initial-assessment)
    benefit = calculateBenefit(interaction-initial-assessment,
    interaction-final-assessment)
    sumOfBenefits = sumOfBenefits + similarity * benefit
  end for
  currentBenefit = sumOfBenefits / numberOfPreviousInteraction
  if bestObject.benefit < currentBenefit then bestObject = currentObject
end for
if bestObject.benefit < 0 then
  bestObject = randomObject
end if
```

Algorithm 1: Collaborative Learning Algorithm

where *w(a,j)* reflects the similarity $\in$ (0,1] between each user j and the active user, a, and κ is a normalizing factor. $\frac{1}{n}$ was used as the value for κ in this work where n is the number of previous users who have interacted with learning object l. *w(a,j)* was set as $\frac{1}{1+\mathit{difference}}$ where difference is calculated by comparing the initial assessment of j and the current-student-assessment, and assigning an absolute value on the difference of the letter grades assigned. So the difference of A+ and B- would be 5 and the difference of D+ and C- would be 1. This is shown as the calculateSimilarity function in Algorithm 1.

v(j,l) is also computed using a difference, not an absolute difference but an actual difference (between the initial and final assessments). For example, *v(j,l)* where j is initially assessed as A+ and finally assessed at B- would be -5, while where j is initially assessed at B- and finally assessed at A+ would be 5. This is shown as the calculateBenefit function in Algorithm 1.

In the absence of other criteria, a user a will be assigned the learning object l that maximizes *p[a,l]*. In the case that the maximum *p[a,l]* is a negative anticipated benefit, a random learning object will be assigned to the user.

It is important to note that for the simulations to validate the Collaborative Learning Algorithm we used a multi-dimensional model of knowledge, described briefly in Section 2.6.4.1. For clarity we will first discuss similarity and benefit in terms of a single dimension; however, this technique would be most appropriately applied in multiple dimensions, with each axis representing a facet of the overall knowledge. Standard hyper-dimensional geometry allows the extrapolation of these examples and is presented in Section 3.2.

3.1 Example

Here we provide a simplified example for illustration. Suppose we track each learning object, LO with [index, [StudentID, initial assessment], [final assessment]]. After multiple interactions with 3 students, S1, S2 and S3, the experiences are shown in Table 3.1.

LO[1; S1(B,C), S3(B,A+)]
LO[2; S1(A,A), S3(C,A-)]
LO[3; S2(B-,A)]
LO[4; S1(C,A-), S2(B,B)]
LO[5; S3(C+,B)]

Table 3.1: Student Experiences Interacting With Learning Objects

At this point the system is slightly positive on the benefit of LO[1] for B students (because one time it raised a student to A+, and another time it lowered a student to C). It is neutral on LO[2] for A students (the lesson didn't change the student's assessment), and very positive for C students (since it raised a C student to an A-). Similarly LO[3] is good for B- students, LO[4] is very good for C students and neutral for B students, and LO[5] is good for C+ students.

Suppose the system were now considering which lesson to recommend for a student, S4, with current-student-assessment of B+. Per Equation 3.1, it would consider LO[1] to have a currentBenefit of $(\frac{1}{1+1} \times \text{-}3 + \frac{1}{1+1} \times 4) \div 2 = \mathbf{0.25}$, LO[2] a currentBenefit of $(\frac{1}{1+2} \times 0 + \frac{1}{1+4} \times 5) \div 2 = \mathbf{0.67}$, LO[3] a currentBenefit of $\frac{1}{1+2} \times 4 = \mathbf{1.33}$, LO[4] a currentBenefit of $(\frac{1}{1+4} \times 5 + \frac{1}{1+1} \times 0) \div 2 = \mathbf{0.5}$ and LO[5] a currentBenefit of $\frac{1}{1+3} \times 2 = \mathbf{0.5}$. In this situation, LO[3] would be recommended. After the system's interaction between S4 and LO[3] there will be more information to reason about with future students. The next B+ student will be assigned to LO[3] if S4 has a positive experience, but will instead be assigned to LO[2] if S4 has a neutral or negative experience with LO[3]. This assumes that no additional students use these learning objects in between S4's interactions.

3.2 Multi-Dimensional Model of Knowledge

For the model of learning employed in the simulations used to demonstrate the value of the Collaborative Learning Algorithm, we wanted a deeper representation than a strict letter grade for a student. Instead of an A- Grade 1 Mathematics student, if we think about a student who excels at recognizing numbers, counting, measurement and chance but struggles with shapes and spatial relations we can consider more complex situations. A possible scenario would be one where experiencing a learning object improves a student's understanding of a very specific part of the curriculum. We model this by representing student knowledge as an array of assessments, each of which corresponds to part of the domain of instruction. The student's overall assessment is then the average of this array. When contrasting the knowledge of two students, it is productive to consider N-dimensional space, where N refers to the number of elements in this array of assessments. The similarity between the students can be thought of as their geometric distance in this vector-based approach.

Consider a high school English course with assessment in "Composition", "Reading and Understanding" and "Spelling and Grammar". Imagine a cube, where each of these skills is one of the axes, and each axis represents the student's current knowledge ranges from 0, complete ignorance, to 1, complete mastery. We can intuitively understand how two students with complete mastery of all 3 skills would be very similar (they would occupy the same vertex of the cube). If we contrast these students with one who has mastered (1) "Reading and Understanding" and "Spelling and Grammar", but is completely ignorant (0) in "Composition" we can see that they would occupy separate vertices of the cube. We can also see how this third student would be more similar to the first two than he would be to a fourth student, who is completely ignorant of all three assessment areas. This is illustrated in Figure 3.1.

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i| \tag{3.2}$$

In order to reason about multi-dimensional knowledge when determining similarity of students for the Collaborative Learning Algorithm, rather than a Euclidean distance, we would need to use Manhattan distance[2] (Equation 3.2) which considers the distance between two points as the sum of the absolute difference between their coordinates. In

[2] Also known as Taxicab geometry, rectilinear distance, L_1 distance, city block distance or Manhattan length [71].

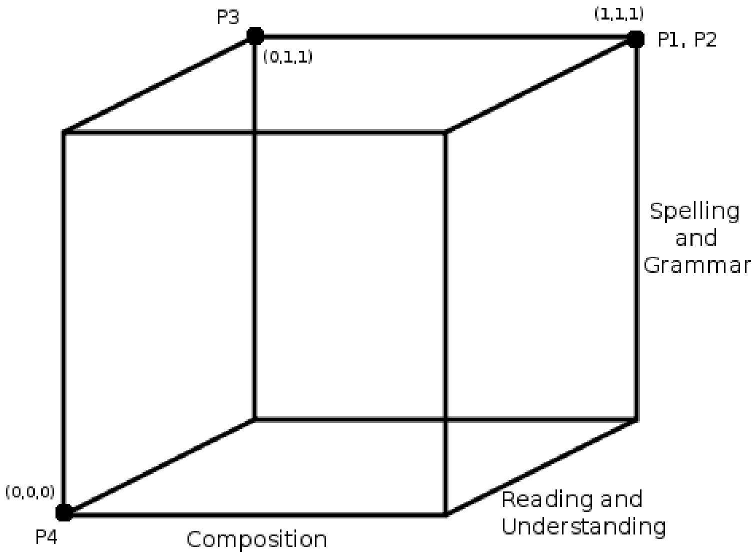


Figure 3.1: Multi-Dimensional Knowledge

this example, the distances between the students P1 (x_1,y_1,z_1) and P4 (x_4,y_4,z_4) would be determined by summing the lengths of projections of the points onto the 3 coordinate axes:

$$\begin{aligned} d &= \sum_{i=1}^{n} |P1_i - P4_i| \\ &= |x_1 - x_4| + |y_1 - y_4| + |z_1 - z_4| \\ &= |1-0| + |1-0| + |1-0| \\ &= 3 \end{aligned} \tag{3.3}$$

The Manhattan distances between the points are displayed in Table 3.2. If we had used Euclidean distance instead, these students would have the distances displayed in Table 3.3. This is because, for the Euclidean distance, distance between two points, for example P1 (x_1,y_1,z_1) and P4 (x_4,y_4,z_4) is determined using the Pythagorean theorem, as illustrated below in Equation 3.4.

	P1	P2	P3	P4
P1	-	0	1	3
P2	0	-	1	3
P3	1	1	-	2
P4	3	3	2	-

Table 3.2: Manhattan Distances Between Points in Figure 3.1

	P1	P2	P3	P4
P1	-	0	1	1.73
P2	0	-	1	1.73
P3	1	1	-	1.41
P4	1.73	1.73	1.41	-

Table 3.3: Euclidean Distances Between Points in Figure 3.1

$$\begin{aligned} d &= \sqrt{\sum_{i=1}^{n}(P1_i - P4_i)^2} \\ &= \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2 + (z_1 - z_4)^2} \\ &= \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} \\ &= \sqrt{1+1+1} \\ &= 1.73 \end{aligned} \tag{3.4}$$

Given that there is an intuitive appeal to considering P1 and P4 as being 3 times more different than P1 and P3 this made the decision to use Manhattan distance appealing[3]. Expanding beyond three dimensions is trivial from a mathematical perspective; however, visualization becomes more difficult.

3.2.1 Multi-Dimensional Example

In Section 3.1 we showed the pre- and post-assessments for each student. For example, S3 went from a C+ to a B, overall. We now consider a student's evaluation to be the average

[3] For future work we may consider alternative measures (see Section 9.2.5.1).

Ability	Before	After
Number Learning	C+	A-
Phonetic Script	C	C+
Spelling Clues/Hidden Words	B-	B-
Words in Sentences	D+	B
Paired Associates	B+	B+
Average	C+	B

Table 3.4: Model of Student P3's Learning

of their evaluation in a number of subtopics in the course of study. Student P3, described in Table 3.4 is a more detailed depiction of Student S3's learning. Consider the 5 separate abilities which are used for the Modern Language Aptitude Test [17]:

Number Learning
This section is designed in part to measure the subjects memory as well as an auditory alertness factor which would affect the participant's auditory comprehension of a foreign language.

Phonetic Script
This section is designed to measure the participant's sound-symbol association ability, which is the ability to learn correlations between a speech sound and written symbols.

Spelling Clues/Hidden Words
This highly speeded section is designed to test the participant's vocabulary knowledge of English as well as his/her sound-symbol association ability.

Words in Sentences
This section is designed to measure the participant's sensitivity to grammatical structure without using any grammatical terminology.

Paired Associates
This section is designed to measure the participant's rote memorization ability, which is a typical component of foreign language learning.

Student P3 above went from a C+ student to a B student overall. If we look at a finer granularity of assessment for her, we might see the following.

Her improvement was not consistent across all abilities, and in fact she showed no improvement at all in two areas ("Spelling Clues/Hidden Words" and "Paired Associates") but a dramatic improvement in two other areas ("Number Learning" and "Words in Sentences"). The overall average is simply the sum of the assessments for each separate ability divided by the number of abilities (5 in this case).

Consider again the specific example in Section 3.1. Below we provide a variation of this example with multiple knowledge dimensions being modeled, for illustration. Suppose we track each learning object, LO with [index, [StudentID, initial assessment] → [final assessment]]. After multiple interactions with 3 students, P1, P2 and P3[4]:

LO[1; P1(A/C/B-/B/B+ → A/D/C/C/C); P3(B/B/C+/B+/B+ → A+/A+/A+/A+/A+)]
LO[2; P1(A/A-/A/A+/A → A/A/A/A/A); P3(D/C/C/B/C → B+/A-/A-/A-/A+)]
LO[3; P2(B-/B-/C+/B-/B → A/A/A/A/A)]
LO[4; P1(C/C/A/D/D → A/B+/A/B+/A-); P2(A/B/B/C/B → A/B/B/C/B)]
LO[5; P3(C+/C+/C+/C+/C+ → B/B/C+/B+/B+)]

At this point the system is negative on the benefit of LO[1] for A/C/B-/B/B+ students and positive on the benefit to B/B/C+/B+/B+ (because one time it raised a student by 4 letter grades, and another time it lowered a student 3 letter grades). A new student who had a B+ assessment for "Paired Associates" would be equally similar to P1 and P3 for LO[1] on this dimension. If this student is more similar to P3 on the other dimensions then the overall recommendation for this student for LO[1] would be slightly positive.

It is neutral on LO[2] for A/A-/A/A+/A students (the lesson didn't change the student's assessment), and very positive for D/C/C/B/C students (since it raised the second student 5 letter grades). Students with lower assessments will tend to be more similar to the second student and therefore are likely to be recommended this learning object, whereas students with high evaluation will be more similar to the first and be less likely to be assigned this learning object. Similarly LO[3] is good for B-/B-/C+/B-/B students, LO[4] is very good for C/C/A/D/D students and neutral for A/B/B/C/B students, and LO[5] is good for C+/C+/C+/C+/C+ students.

Suppose the system were now considering which lesson to recommend for a student, P4, with current-student-assessment of B+/B/A-/B-/A (overall a B+ as is the case with the example in Section 3.1). The similarity would now be calculated looking at each of the individual abilities rather than comparing the overall assessment. For example, the similarity between P4 and the P1 experience attached to LO1 would be:

[4] As was the case earlier, these are each more detailed versions of S1, S2 and S3.

$$\begin{aligned}
sim(P1,P4) &= \frac{1}{1+difference(P1,P4)} \\
&= \frac{1}{1+\sum_{d\in dimensions}|assessment_d(P1)-assessment_d(P4)|} \\
&= \frac{1}{1+|A-B^{+}|+|C-B|+|B^{-}-A^{-}|+|B-B^{-}|+|B^{+}-A|} \\
&= \frac{1}{1+|2|+|-3|+|-3|+|1|+|-2|} \\
&= \frac{1}{1+2+3+3+1+2} \\
&= \frac{1}{12} \\
&= 0.0833 \qquad (3.5)
\end{aligned}$$

In the same way, the various similarities would be calculated:

Student Experience	Similarity with P4
P1 on LO1	0.0833
P3 on LO1	0.100
P1 on LO2	0.0910
P3 on LO2	0.0435
P2 on LO3	0.0910
P1 on LO4	0.0435
P2 on LO4	0.100
P3 on LO5	0.0625

In accordance with Equation 3.1, LO[1] would have a currentBenefit of (0.0833 × -3 + 0.1 ×4) ÷ 2 = **0.075**, LO[2] a currentBenefit of (0.091 × 0 + 0.0435 ×5)÷ 2 = **0.154**, LO[3] a currentBenefit of 0.091 × 4 = **0.364**, LO[4] a currentBenefit of (0.0435 × 5 + 0.1 ×0)÷ 2 = **0.109** and LO[5] a currentBenefit of 0.0625 × 2 = **0.125**. Even in this fine grained situation, LO[3] would still be recommended. After the system's interaction between P4 and LO[3] there will be more information to reason about with future students. The next student who is similar to P4 may be assigned to LO[3] if P4 has a positive or neutral experience, but may instead be assigned to LO[2] or LO[5] if P4 has a negative experience with LO[3]. This assumes that no additional students use these learning objects in between P4's interactions.

In our study with human users, when assessing similarities of students we also used a multi-dimensional model of knowledge. This is discussed in Chapter 7 with details provided in Appendix E.

3.3 Simulation

We used simulated students to validate our content sequencing approach. Our motivation for performing this simulation was to validate that, in the experimental context, our approach leads to a higher average learning by the group of students than competing approaches.

In order to simulate the learning achieved by students, the following approach was used. In this discussion below, for clarity, we discuss learning objects with just 1 overall knowledge dimension. Let *LOK[l,k]* represent some learning object *l*'s target instruction level of knowledge *k*, such that *LOK[l,k]* $\in$ [0,1] where 0 is complete ignorance and 1 is complete mastery within the course of study. For example, the target instruction level might be 0.68 for a 90 minute lab on recursion, since students have completed previous learning but are still gaining an understanding of nuances.

Learning objects also have an impact, which can be positive or negative[5]. Let *I[l,k]* $\in \mathcal{R}$, represent the impact of learning from learning object *l* on the knowledge *k*, that is, in the optimal case how much the learning object can adjust a student's knowledge *k*. The impact can be thought of as, for a student at the target level, what is the expected learning benefit of the object. This is information used by our approach to simulate the learning that is occurring.

Let *UK[j,k]* represent user *j*'s knowledge of *k* $\in$ K, such that *UK[j,k]* $\in$ [0,1]. An example from computer science would be a knowledge of recursion recorded to be at 0.33. This would be interpreted as the student has an understanding of 33% of the course content dealing with recursion. The cardinality of K would be the number of dimensions making up the knowledge domain being simulated.

After an interaction with an object *l*, a user *j*'s knowledge of *k* is changed by:

[5] The negative impact was introduced to simulate the possibility of misinformation from a poor quality learning object or a learning object that does a good job teaching one concept, while undermining the understanding of another concept.

$$\Delta\mathit{UK[j,k]} = \frac{I[l,k]}{1 + (UK[j,k] - LOK[l,k])^2} \tag{3.6}$$

This has the implication that the impact of a lesson is at a maximum when the student's knowledge level matches the target level of the learning object. As the two values differ, the impact of the lesson exponentially decreases.

Based on this change, the user's knowledge in that area is updated as:

$$UK'[j,k] = UK[j,k] + \Delta UK[j,k] \tag{3.7}$$

The user's average knowledge can then be calculated as:

$$\overline{UK}[j] = \frac{1}{|K|}\sum_{k \in K} UK[j,k] \tag{3.8}$$

3.4 Simulation Parameters

For our experiment, variable numbers of simulated students and learning objects were allowed to interact over a set number of trials, chosen as 100[6]. Each simulated student was randomly assigned values for each of 6 knowledges, each evenly distributed in the range [0,1]. A multi-dimensional structure for knowledge was used to ensure randomly generated students were distinct from one another, and to provide a rich model for simulated learning. In a real world context, this can be thought of as students who have a better understanding of one part of a course of study compared to another part of the same course. Each learning object was randomly assigned a value for the target level of instruction for each knowledge, evenly distributed in the range [0,1]. Impact values were assigned to learning object knowledges, randomly and evenly distributed in the range [-0.05, 0.05]. The values of -0.05 and 0.05 were chosen such that no single learning object could radically change a user's knowledge level (at most it can adjust it by 5%).

Each experimental condition was repeated for 20 iterations, and the mean of the average knowledges of all students after each trial was graphed (see Section 3.5). In this context, the average knowledge might be thought of as: if a final mark for the CS 101 course was to be assigned to a student based on their current understanding of the course content,

[6] Simulations with other runtimes are detailed later in this section.

what would it be? A student's final mark will be based on their knowledge of a number of areas, such as introductory data structures, recursion, sorting, introductory proofs, and programming in a specific language.[7] In order to explore the value of our approach, we graph the performance of several algorithms to select a learning object for each student, over a number of trials. After each trial, the average knowledge under each condition is compared.

In order to plot learning curves, the average knowledge ($\in [0, 1]$) of all students is plotted against their progress in the course of study. Algorithms perform well when the average knowledge attained by students is high. A set of algorithms to select learning objects for students were run, to demonstrate the value of the proposed approach.

Two reference points were created to compare our approaches against.

Random Association: One reference was created by associating each student with a randomly assigned learning object in each trial. Given that any intelligent approach to matching students with objects should outperform blind chance, this was viewed as the lower limit.

Greedy God: The other reference point, the greedy god reference point, was created by giving the algorithm full access to the fine-grained knowledge levels of the students and learning object, testing what the outcome would be for every possible interaction, then choosing the best interaction for each student for each trial. The results, based on an omniscience not typically available in real world learning environments, was viewed as a ceiling on the possible learning benefit of any approach. This curve thus represents "the ideal".

Three variations of Algorithm 1 were then run. The impact values and target levels of objects are used for the reasoning of the greedy god algorithm. In contrast, for the following three algorithms, our ecological approach is used to select the learning objects to be presented to users (so based on their similarity to previous students who have experienced these objects and on the benefit that these students derived). We assume that as each simulated student is assigned a learning object, that student's interaction with the object can be used as the "previous experience" to which subsequent students are matched. Using our ecological approach, learning objects presented to students should end up being ones that have an effective combination of impact value and target level (i.e. beneficial to those previous users and at a somewhat similar level of knowledge).[8]

[7] We make the simplifying assumption that all knowledges contribute equally to this progress assessment and calculate it as the average knowledge. We discuss this further in Section 9.2.5.1.

[8] We assume that a learning object can be experienced more than once by the same student. Further motivation for this assumption and discussion of alternative models is presented in Section 9.2.2.3

Raw Ecological: For the raw ecological approach, each student is matched with the learning object best predicted to benefit her knowledge. 3 trials were run where each student was randomly assigned to a learning object. For the remaining 97 trials, the algorithm matched each student with the learning object best predicted to benefit her knowledge[9]. The initial three trials with random associations provided rough information about the learning objects and students, which the algorithm used and refined over the course of the remaining trials.

Pilot Group: For the pilot group ecological approach, the algorithm assigned a subset of the students (10%) as a pilot group. The algorithm systematically assigned students in the pilot group 100 learning objects in sequence, to explore those learning objects' impact. The remaining 90% of the students used the effects of these initial interactions, in conjunction with their own experience gathered through separate interactions, to reason about the best sequence.

Simulated Annealing: Our third ecological approach was inspired by simulated annealing, which in turn was inspired by the metallurgical approach of heating and cooling to induce change in a material. For this approach, we had a "cooling" period, which was the first 1/2 of the trials. During this period, for every student, there was an inverse chance, based on the progress of the trials, that they would be randomly associated with a lesson; otherwise, the ecological approach would be applied. For example, in the first trial, every student would be randomly associated with a learning object, but by the 25th trial, each lesson would have a 50% chance of being randomly associated. After the cooling period was over (the 50th trial), every student was repeatedly assigned to a learning object by ecological reasoning.

3.5 Results

As seen in Figure 3.2 the random associations of students with learning objects is clearly and consistently shown to be an inferior approach[10] to improving the average knowledge of a group of students, as expected. Similarly, an omniscient sequencer using perfect

[9] Since each student's knowledge is now multi-dimensional the difference calculated to determine benefit is now a sum of the differences of each knowledge dimension. The numeric values for knowledge are converted to one of the concrete letter grade levels before performing the computation.

[10] Because there was an even mixture of learning objects which improve or degrade the students' knowledge, when learning objects are randomly assigned the overall average knowledge tends to stay the same, leading to the flat curve seen in the graphs in this chapter.

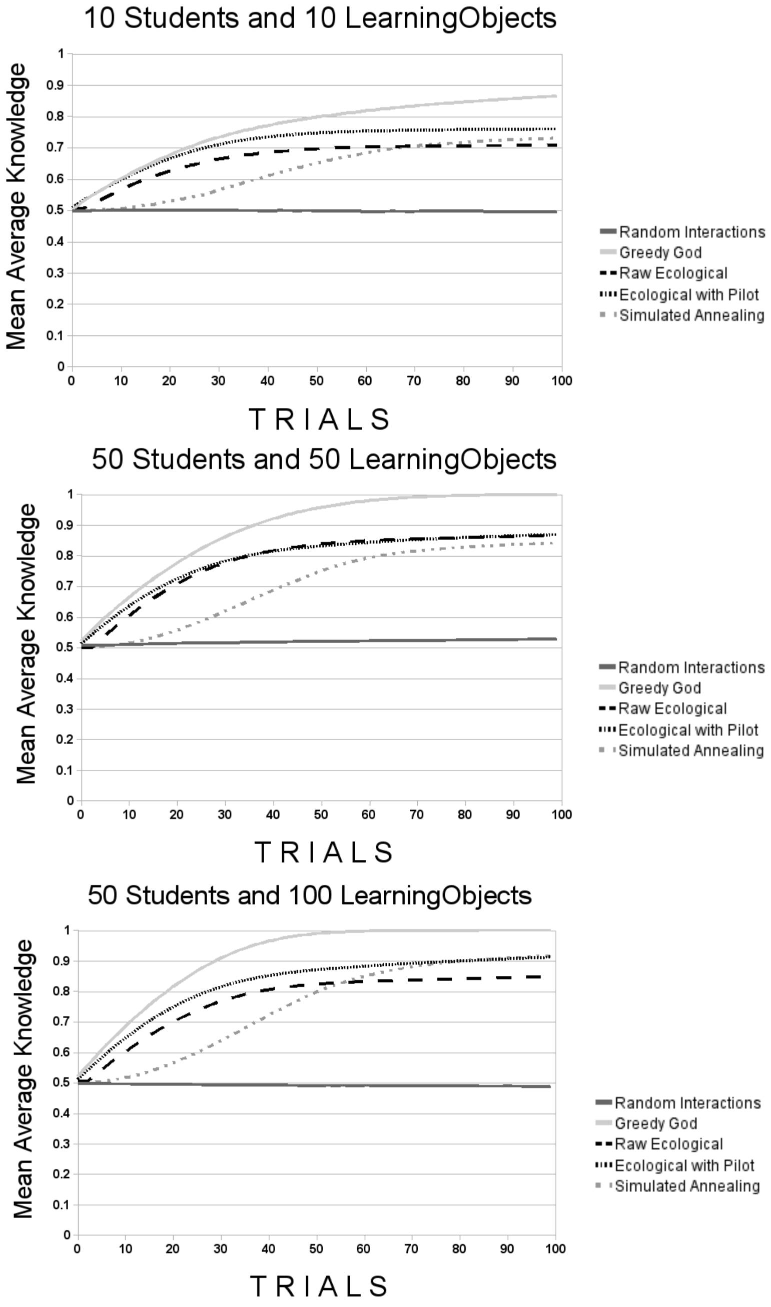


Figure 3.2: Comparison of 5 Approaches to Sequencing for Vary Numbers of Students and Learning Objects

knowledge of students, learning objects and the outcome of a potential interaction (greedy god) can consistently produce the greatest learning benefit.

Contrasting our ecological techniques (which would each be feasible in a real educational setting) with these reference points, provides illumination on the usefulness of the ecological approach in this setting. Reasoning intelligently, in this manner, has produced greater knowledge in a shorter number of trials for the group of students as a whole compared to a random association.

As asserted by his paper of the ecological approach [54], we see McCalla's predicted impact that with more learners the ecological approach's performance improves. A correlation of improved performance with an increase in number of learning objects was also seen. This makes intuitive sense: if an intelligent tutoring system (ITS) is given a larger repository of learning objects to assign, we would expect it to be able to find objects better suited to a particular student.

While Figure 3.2 seems to show superior performance of the ecological approach with a pilot group, it is important to remember that 10% of the class was used as a pilot group for this experimental condition. These were not included in the average assessed knowledge. Their increased knowledge, which would be roughly equivalent to the lack of increase shown by random associations, is omitted and the improved performance of the remaining students can be viewed as at the expense of the pilot group.

The "Simulated Annealing" technique was interesting as it underperformed the other two techniques during its "cooling period" but quickly gained ground after the cooling period was complete. This is due to the randomness added during the cooling period leading to a greater exploration of the possible interactions between learning objects and students. This improved understanding of the two groups could then be used when reasoning about which students to match with which learning objects in later trials. In the largest condition (50 students and 100 learning objects), simulated annealing matched the performance of the pilot group condition, without sacrificing 10% of the class. With the correct choice of cooling periods, this technique shows promise for delivering comparable long term performance at the expense of early progress for the entire group instead of no progress for a pilot group.

3.6 Error

Algorithm 1 proposes the selection of learning objects for students based on their similarity to peers and the benefits these peers have obtained from existing learning objects. Both similarity and benefit are determined in terms of the assessment levels of the students (obtained by pre- and post-test of each student, mapped to a level of a letter grade).

Since assessment in the real world is both imprecise and occasionally inaccurate[11], there is merit in exploring how well the algorithm would perform when validated in a simulated environment where the assessments include an element of error. Our interest is in how well the algorithm makes recommendations using this noisy data[12].

3.6.1 Approach

Our original hypothesis was that we expected errors in assessment, of the form of introduced noise, to degrade the learning curves observed. That is, we expected that as the noise increased, the slope of each learning curve embodying Algorithm 1 would decrease and they would gradually move away from the ideal greedy god curve and towards the random baseline. It was expected that rather than converging on perfect knowledge (the 1 value on the y axis), the curves would converge on a lower value (that would drop ever lower as the noise increased). This would happen as the number of interactions between students and learning objects increased (modeled as trials in the x axis).

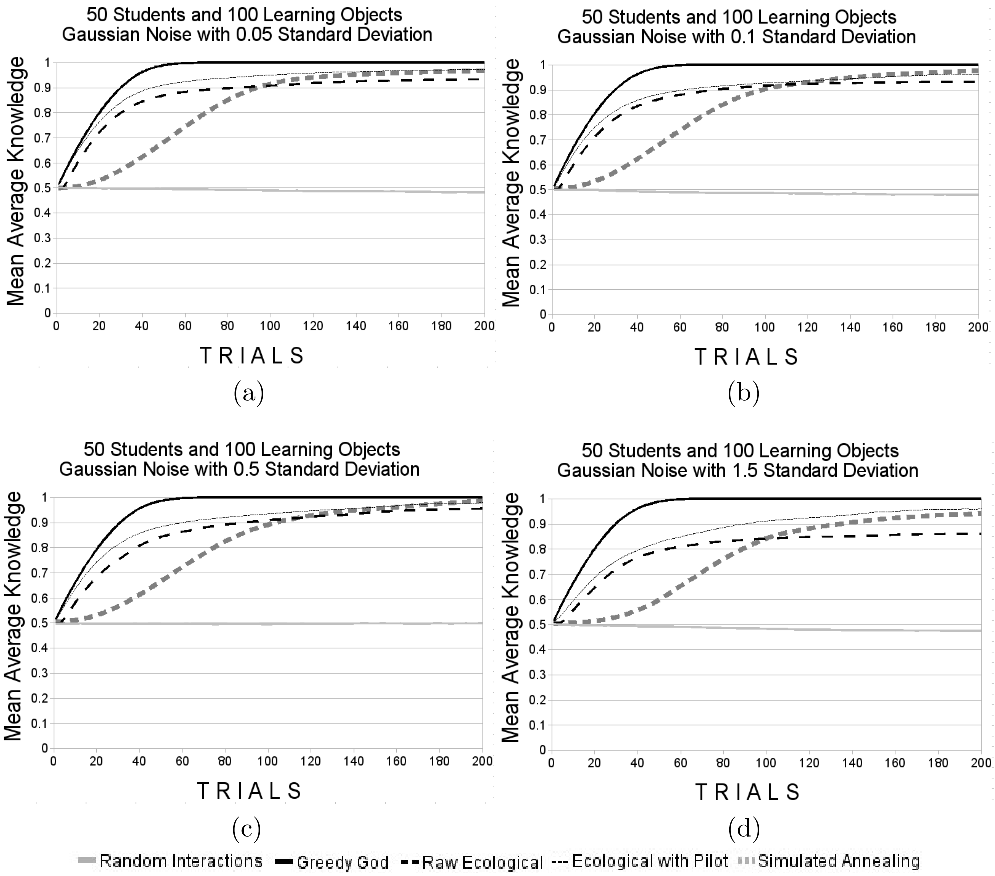
If the algorithm were robust in the face of error (that is, if the learning curves stay closer to the ideal case), this tells us that our approach can handle errors in assessment and continue to provide worthwhile recommendations to students, even in the face of assessment errors. Poor performance, where the slope of the curves would drop quickly towards the random baseline, would tell us that this approach is highly dependent on good assessments (and would thus constrain the environment where it would be appropriate to use this approach).

In order to produce the error, we modified the assessment function in our simulation. Rather than mapping a knowledge level (continuous values in the range [0,1]) to a discrete level {A+, A, ... F, F-}, we first added a random number, using a Gaussian distribution, with a mean of zero and a standard deviation of 0.05, 0.1, and 0.5[13]. These experiments,

[11] Even with an ideal assessment tool, there will still be situations where students mistakenly give incorrect information that they understand (known as a slip) or accidentally give the right answer to something they don't understand (known as a guess).

[12] It is important to note that there are different approaches to model a "bad assessment". By randomly adding noise, we are modeling an assessment that has variability in every assessment. This does not model an assessment with a systemic bias, for example, one that always evaluates C+ students as D students.

[13] The idea is that if a student could be modelled with an erroneous assessment level (e.g. B vs. A) then with greater standard deviation, the likelihood of an erroneous label increases. Note that values closer to the true value will still be the most likely to be assigned.



(a)

(b)

(c)

(d)

Figure 3.3: Comparison for Varying Standard Deviation of Error in Assessment

taken as a whole, should provide us with an understanding of how increasing levels of noise in the assessment affects the effectiveness of curriculum sequencing performed by this approach. The greedy god and random baselines remained unchanged, since neither relied on assessment.

3.6.2 Results

We did not see evidence of what we originally expected with the 3 graphs created with standard deviation (0.05, 0.1, 0.5). Instead, all 3 curves looked quite similar to the learning curves obtained using this approach on data without noise added to it, as presented in

Section 3.5. In Figure 3.3 (a)(b)(c), all three variations of the algorithm are performing well, in getting close to attaining the ideal average level of knowledge for students (i.e. the greedy god) by the end of the 200 trials. Note, as expected, simulated annealing takes longer to converge, as it is coping with random information at the beginning.

An initial concern was that our experiment might somehow be accidentally determining the appropriate learning objects without relying on the assessment. To test this concern, we replaced the assessment function with a function that randomly provides one of the 18 discrete levels (instead of an assessment, it provides a random grade). Since the three variations on this approach (ecological, ecological with pilot and simulated annealing) all rely on assessments to function, our expectation was that this change would produce 3 curves that were degraded to the performance of the random baseline. This is the result we saw (see Section 3.8).

In all, these results tell us that this approach is, in fact, highly robust with noisy data. As long as there is a tendency for an assessment to be closer to a correct value than an incorrect value, this approach will steadily improve the curriculum sequence suggestions as more data is obtained. Realistic amounts of noise, which are expected with any assessment, would seem to be acceptable to the functioning of this approach.

What is happening with our approach is the following. Suppose the error in assessment led to an inappropriate learning object being proposed for a new student (e.g. the previous student was assessed as deriving benefit from that object where, in fact, he had not). The simulation would model this new student's interaction with the learning object. Now, this should reflect a poor increase in knowledge since the learning object recommended is a poor match for this student (i.e. the student's knowledge level is not attuned to the target level of knowledge (Equation 3.6)). When the new student's assessment is modeled (and will likely show poor learning gains), it will then be attached as one of the experiences with that learning object. As a result, this learning object would now be less likely to be assigned to students in the future.

Part of the power of this peer-based algorithm is the ability to correct mistakes. If a recommendation is made because of an inaccurate assessment (that is, an interaction that was actually harmful is instead recorded as being useful) this will lead to further recommendations of that object to similar students. However, when those similar students use the object, they will in turn be assessed. As this approach considers all previous interactions, with more students interacting with the object, a larger history of interactions will accumulate. The average of these assessments will approach the true value, even if some of those assessments are distorted by noise. As this happens, the system is less likely to recommend the bad object, and will increasingly direct students to a better choice, leading to a self-correcting system.

While our initial feeling was that a standard deviation of 0.5 was a large amount of noise, we then ran an experiment with a standard deviation of 1.5 (see Figure 3.3 (d)). The consequence of this is that we're adding noise which is very likely to move data points anywhere in the range (with a standard deviation of 1.5, there is roughly a 25% chance of a perfect knowledge of 1 being mapped to a F- or, conversely of a complete absence of knowledge of 0 being mapped to an A+). With this massive amount of noise being added, we then saw the degradation we had initially expected (with the ecological condition converging on 0.8 instead of 1.0).

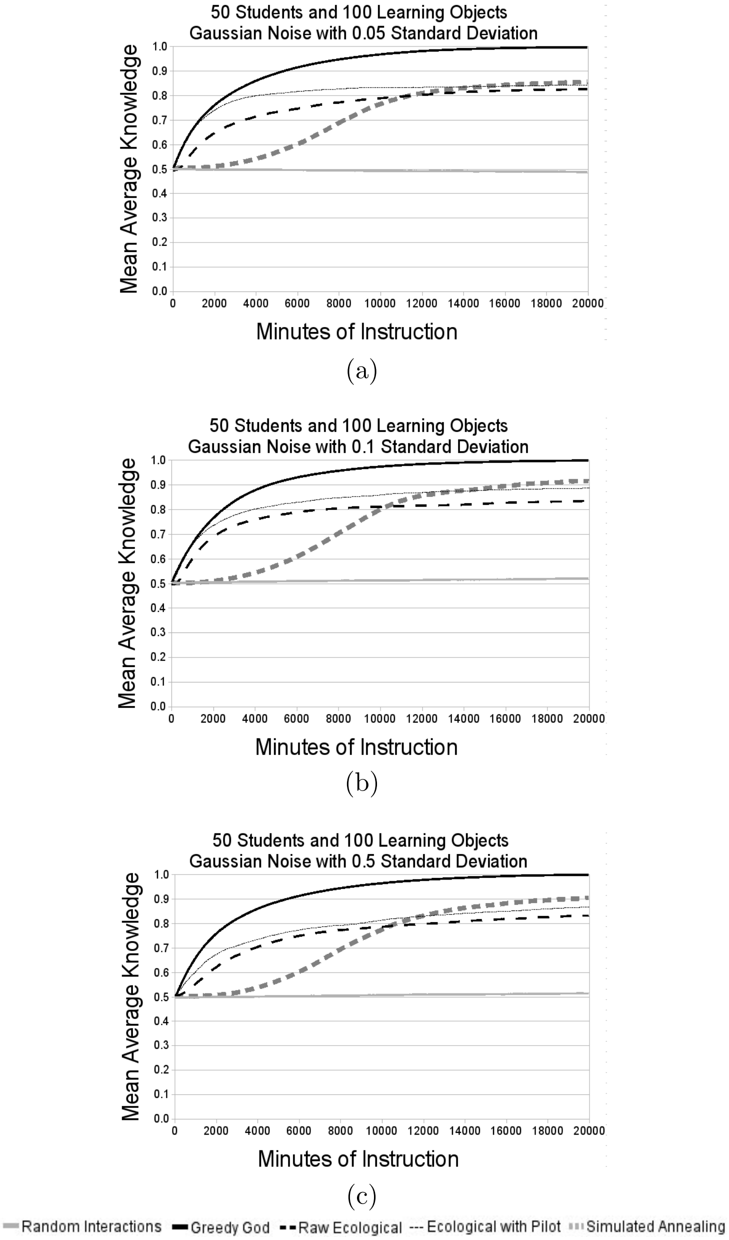
3.7 Variable Time of Instruction

In the previous approach outlined in this chapter, which learning object should be assigned to a particular student is dependent on similarity of peers and the previous learning benefit obtained by those peers, alone. We explored a new extension, where we incorporate reasoning about the length of time it takes to complete an interaction with a learning object as well.

Clearly, in real learning situations, learning events can take variable amounts of time. Watching a recording of a lecture might take 76 minutes, while attending a day long seminar might take 8 hours. Rather than making the simplifying assumption that each interaction with a learning object will take an equivalent length of time, we can incorporate this concept into our reasoning.

calculateBenefit in Line 8 of Algorithm 1 then needs to be modified to incorporate time. Rather than consider the benefit of the learning object, we can think of the proportionate benefit, that is, how much benefit it provides per minute of instruction (assuming a repository where each learning object's average time to completion is recorded). This can be calculated by dividing the benefit of the learning object by the length of time it takes to complete the interaction for the average student.

We are interested in ensuring that, with this more sophisticated consideration incorporated, the approach outlined in Algorithm 1 continues to provide worthwhile recommendations for curriculum sequencing.



(a)

(b)

(c)

Figure 3.4: Comparison for Varying Standard Deviation of Error Including Time of Instruction

3.7.1 Approach

We modified the previous approach (Section 3.3 and 3.6.1) such that, as well as generating a random set of target instruction levels for each learning object, we also generated a random length of completion (ranging from 30 to 480 minutes). We used 50 students, 100 learning object and three runs – an error of 0.05, 0.1 and 0.5 standard deviation – each time for 20000 minutes of simulated instruction. As well as the random and greedy god baselines, we again considered the raw ecological, ecological with pilot and simulated annealling variants. These results are displayed in Figure 3.4 (a), (b) and (c).

It is worthwhile to note that initially we experimented with about 2400 minutes of instruction, based on this being roughly the amount of instruction in a typical university course. This was determined to be far too short a length of experiment as the learning curves reflected only the initial part of the graphs shown here. Our conclusion was that we were simply failing to see, yet, the benefit to learning that the students achieve and that either longer lesson times were needed or that it may be valuable to track students over multiple classes.[14]

3.7.2 Results

With the increased time provided in Figure 3.4, we did indeed attain the kind of student learning that we expected (reasonably high average level of knowledge, for students). With the added sophistication of allowing learning objects to require different lengths of times to complete, this approach continues to make worthwhile curriculum recommendations to students. The fact that all three variants on the algorithm are approaching the ideal of the greedy god at the end of the trials is encouraging. As expected, the greater the standard deviation of error introduced, the more challenged each algorithm is to attain appropriate student knowledge levels, but the differences between Figure 3.4 (a) (b) and (c) are still relatively minor.

An unusual feature is that the learning curves approach a final knowledge less than 1, whereas in the experiments of Figure 1 they approached one. Initially this was thought to be a consequence of the introduced error; however, considering the curves from the error section above does not support this idea. It is possible that each approach is developing a bias towards short lessons, and is therefore not taking advantage of the full range of learning objects that may help the students approach complete mastery.

[14] Although we use the units minutes, these are best thought of as unspecified time units, since no effort was spent trying to calibrate whether the educational gains were appropriate for the length of instruction.

Historically, learning gain has been the accepted metric for measuring a learning event for ITS researchers. One alternative which is being considered is to use the proportional learning gains [37], which is defined as: $\frac{\text{post-test}-\text{pretest}}{1-\text{pretest}}$. This would be a useful alternative for avoiding a bias towards interactions where a student has a low initial score, if this formula were used instead on the right hand side of the equation in line 8 of Algorithm 1. For example, with this metric, advancing from A to A+ is a greater gain than advancing from B to B+. We explore the use of this metric in our human study, discussed in Chapter 7.

3.8 Random Assessment

The software developed to run the simulations in this work eventually amounted to thousands of lines of source code. While the code was developed carefully and tested during development, bugs are inevitable in a codebase of this size. One of the techniques we performed to validate the source code during development was to run simulations that were different from what the code was written for, ensuring that the results matched the expectation.

One example of this was a simulation where the assessment portion of the source code was replaced with a method that would return a random grade instead. The expectation was tha this would degrade the performance of all techniques to match the random baseline. We expected this since our techniques use assessment data to reason about student benefit and similarity to other students, and without this data it cannot provide reasonable recommendations.

The results obtained from this experiment are shown in Figure 3.5 and they match what was expected.

3.9 Large Corpus

It has been suggested [54] that ecological approaches, rather than degrading with large amounts of information, improve. Intuitively this makes sense, with more data better recommendations should be possible. In order to investigate this, we considered a student group interacting with a large library of learning objects (5000 objects). Collecting a massive amount of educational content offers a valuable resource, but also introduces the challenge of navigating a large corpus.

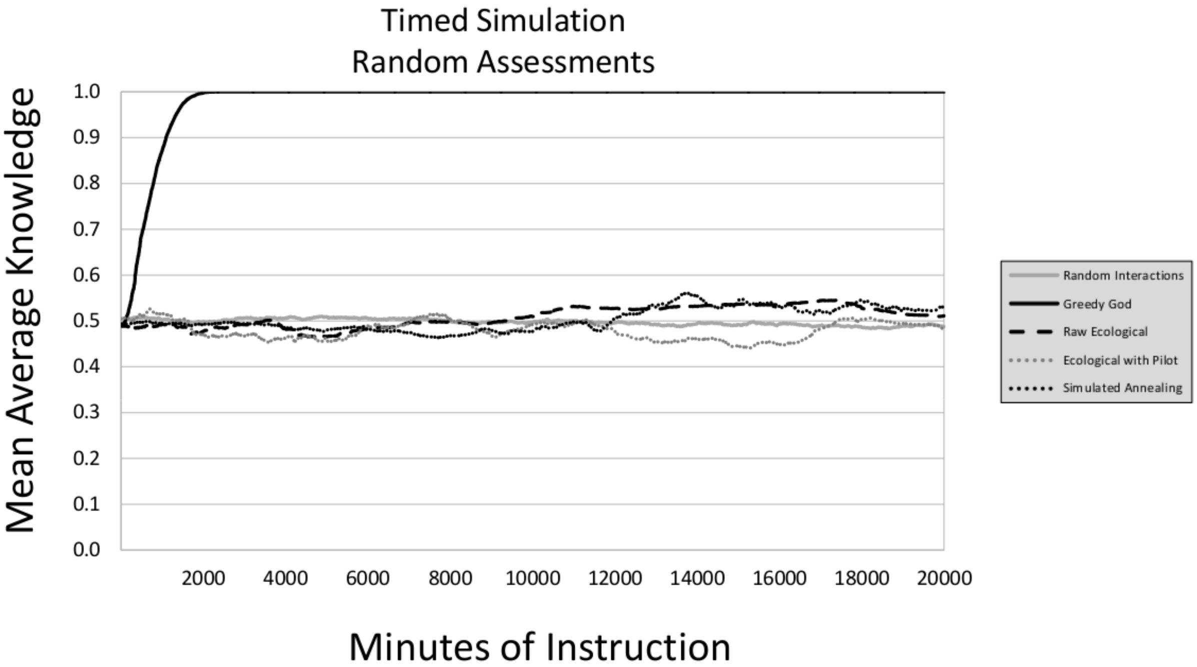


Figure 3.5: Effect of Random Assessment in Place of Student Assessments

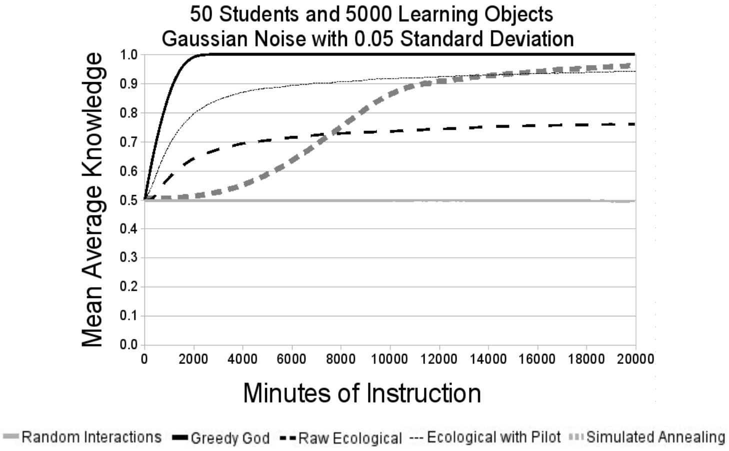


Figure 3.6: Error and Time of Instruction for a Large Corpus

When we consider a simulation with dramatically more learning objects (Figure 3.6), we see that both the simulated annealing and the ecological with pilot learning curves become steeper. This corresponds with McCalla's prediction. The ecological with pilot group has a sustained improvement and outperforms the raw ecological more dramatically than in previous experiments. Similarly, the simulated annealing conditions performs well with a larger library. The additional exploration of the corpus available to these algorithms in their initial phases appears to be providing some valuable benefit. Note that these curves approach the ideal average knowledge of the greedy god to a greater extent with the larger repository (compared to the small repository used in Figure 3.4), which again provides support for McCalla's hypothesis.

Chapter 4

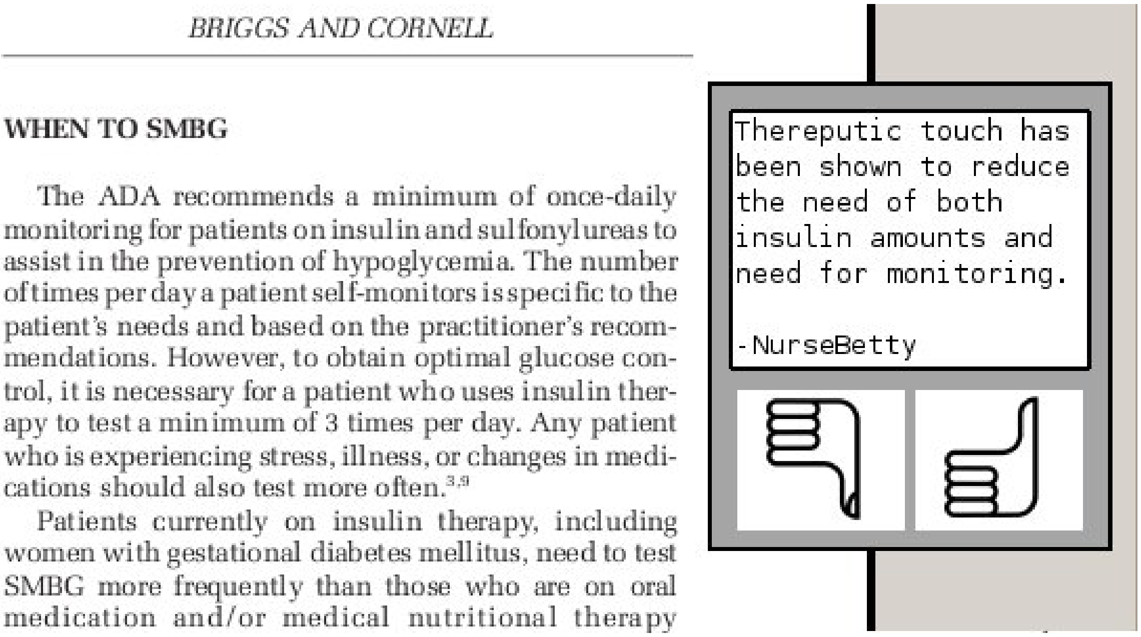
Annotations

Given a set of annotations, short text messages left by previous students, that can be attached to a particular learning object a student is experiencing, which annotations should be shown to a student in order to improve their learning? Our approach creates personalized recommendations of annotations for students, taking into account the preferences of peers, incorporating both similarity and reputability.

4.1 Motivation and Overview

In order to understand our intended use of annotations in peer-based intelligent tutoring, we offer a few examples. The learning object presented in Figure 4.1 is for tutoring caregivers in the context of home healthcare (an application area in which we are currently projecting our research [63]). The specific topic featured in this example is the management of insulin for patients with diabetes. This annotation recommends therapeutic touch (a holistic treatment that has been scientifically debunked but remains popular with nurse practitioners). It would detract from learning if presented and should be shown to as few students as possible.

Consider now: *In an ITS devoted to training homecare and hospice nurses, one section of the material discusses diet and how it is important to maintain proper nutrition, even for terminal patients who often have cravings for foods that will do them harm. One nurse, Alex, posts an annotation saying how in his experience often compassion takes higher precedence than strictly prolonging every minute of the patient's life, and provides details about how he has discussed this with the patients, their families and his supervisor.*



BRIGGS AND CORNELL

WHEN TO SMBG

The ADA recommends a minimum of once-daily monitoring for patients on insulin and sulfonylureas to assist in the prevention of hypoglycemia. The number of times per day a patient self-monitors is specific to the patient's needs and based on the practitioner's recommendations. However, to obtain optimal glucose control, it is necessary for a patient who uses insulin therapy to test a minimum of 3 times per day. Any patient who is experiencing stress, illness, or changes in medications should also test more often.[3,9]

Patients currently on insulin therapy, including women with gestational diabetes mellitus, need to test SMBG more frequently than those who are on oral medication and/or medical nutritional therapy

Thereputic touch has been shown to reduce the need of both insulin amounts and need for monitoring.

-NurseBetty

👎 👍

Figure 4.1: Example of a low-quality annotation [9]

As a real world example of how the material was applied, and the introduction of higher ethical reasoning beyond the standard instruction, this is a very worthwhile annotation to show to other students. Beyond helping students gain a better understanding of the material, this can also assist them in their motivation as they connect the material to real, on-the-job activities they will need to perform in the future.

Two specific challenges, however, arise and need to be addressed when deciding which annotation to show include:

a) some annotations are effective for certain students (but not others)

b) some annotations might appear to be undesirable but in fact lead to learning benefits.

We briefly outline these two cases, below.

Consider now: *A section on techniques for use with patients recovering from eye surgery in a home healthcare environment has some specific, step-by-step techniques for tasks such as washing out the eye with disinfected water. A nurse, Riley, posts an advanced, detailed* *comment about the anatomy of the eye, the parts that are commonly damaged, a link to a medical textbook providing additional details and how this information is often of interest to recovering patients. The remedial students struggling with the basic materials find this annotation overwhelming and consistently give the annotation bad ratings, while advanced students find this an engaging comment that enhances the material for them and give it a good rating.*

As will be seen, since our approach reasons about the similarity of students, over time, this annotation will be shown to advanced students, but not to students struggling with the material.

Some annotations might appear to be undesirable but in fact do lead to educational benefit and should therefore be shown. Consider now: *An annotation is left in a basic science section of the material arguing against an assertion in the text about temperatures saying that in some conditions boiling water freezes faster than cooler water. This immediately prompts negative ratings and follow-up annotations denouncing the original annotator to be advocating pseudo-science. In fact, this is upheld in science (referred to as the Mpemba effect). A student adds an annotation urging others to follow a link to additional information and follow-up annotations confirm that the value of the original comment that was attached.*

While, at first glance, the original annotation appeared to be detracting, in fact it embodied and led to a deeper, more sophisticated understanding of the material. As will be seen, our approach focuses on the value to learning derived from annotations and thus supports the presentation of this annotation.

4.2 Supporting Annotations of Web Objects

Our proposal is to allow peers to leave commentary on web objects as well as learning objects which future peers may then be viewing, to increase their knowledge. In order to make effective decisions about which annotations to show to each new user, we require a bootstrapping phase. Here, a set of peers will be invited to leave commentary on objects, another set of peers will be exposed to all of these commentaries, leaving a thumbs-up or thumbs-down approval rating and then once this phase is complete, we will have an initial set of annotations with attached approval ratings.

Once the bootstrapping phase is over, we are now able to reason about which annotations to show a future user based on

- the reputation of the annotation (how many thumbs up vs thumbs down)
- the reputability of the annotator (overall all annotations left by this user, what percentage received thumbs up)
- the similarity of the peers in terms of how they have rated the same annotations in the past

Our proposed formulae for modeling reputation and similarity and our proposed algorithm for determining which annotations to show to each new user are outlined in detail below.

4.2.1 Overview of the Reasoning

Ultimately, a user should be presented those annotations with the highest overall personalized reputation. The algorithms below outline the interplay between the required calculations that form the backdrop of this decision making.

```
{Consider user as an annotator}
if number of annotations by user == 0 then
  Reputation(user) = 0.5
else
  Reputation(user) = 0
  for each annotation $a$ created by user do
    Reputation(user) += calculateAnnotationReputation($a$){Algorithm 8}
  end for
  Reputation(user) /= number of annotations by user
end if
Return: Reputation(user)$\in [0, 1]$
```

Algorithm 2: User Reputation

Our proposed model for reasoning about which annotations to show a new user u integrates: (i) the annotation's initial reputation (equal to the reputation of the annotator, as calculated in Algorithm 2 - in turn based on how much his previous annotations were liked) (ii) the current number of votes for and against the annotation, adjusted by the similarity of the rater with the user u (calculated using Algorithm 3) to value votes by similar users more highly. The reputation of each annotation for a user u is calculated by

```
similarity (User c, User r)
votesSame = 0
votesDifferent = 0
for each annotation voted by both do
  if current.vote == rater.vote then
    votesSame += 1
  else
    votesDifferent += 1
  end if
end for
similarity = $(\text{votesSame} - \text{votesDifferent})/(\text{votesSame} + \text{votesDifferent})$
return similarity$\in [-1, 1]$
```

Algorithm 3: User Similarity

Algorithm 4, which uses one of three approaches in order to determine the annotations with the highest reputation which should be shown.

Note that we model the overall reputation of a student, initially set as 0.5 on a scale of 0 to 1, where 1 is the most reputable, and afterwards based on the extent to which the student's previous annotations have been found useful by other students (in Algorithm 2). Note as well that students' ratings for or against the annotation serve to adjust the overall reputation of the annotation (as in Algorithm 4) and indirectly, the annotation author's reputation (as in Algorithm 2). It is clear as well that once a student has provided a set of ratings on annotations, it is then possible to reason about their similarity to other students, based on mutually rated annotations (as in Algorithm 3). This allows the probability that an annotation is shown to be determined both by the overall quality of that annotation as rated by all students and by an adjustment due to the similarity between the current student and the students who have previously rated the annotation (as in Algorithm 3)[1].

[1] For the Mpemba effect mentioned earlier, this similarity weighting can allow the revival of an initially suppressed annotation. Imagine that a rater who understands the Mpemba effect happens to see this mostly suppressed annotation. After he gives it a thumbs up, it is now more likely to be seen by similar students who may also give it a thumbs up. Eventually, a community of students who can consider and understand the nuances of this phenomena can promote this annotation to one another while the student body as a whole continues to suppress it.

```
calculateAnnotationReputationSpecific (Annotation a, User u)
if a has no votes then
   reputation = a.initRep{Return the reputation of the annotator at the time the
   annotation was created}
else
   for each vote on annotation do
      sim = similarity(u, voterUser){Algorithm 3}
      if vote.for then
         votesFor += 1 + 1 * sim
      else
         votesAgainst += 1 + 1 * sim
      end if
   end for
   if using tally for reputation then
      reputation = calculateAnnotationReputationSpecificTally(votesFor, votesAgainst,
      annotation.initRep){Algorithm 5} ‘
   else if using trust-based for reputation then
      reputation = calculateAnnotationReputationSpecificTrustBased(votesFor,
      votesAgainst, annotation.initRep){Algorithm 6}
   else
      {using Cauchy for reputation}
      reputation = calculateAnnotationReputationSpecificCauchy(votesFor,
      votesAgainst, annotation.initRep){Algorithm 7}
   end if
end if
return reputation $\in [0, 1]$
```

Algorithm 4: Annotation Reputation

```
calculateAnnotationReputationSpecificTally (real votesFor, real votesAgainst, real
initRep)
tally = (votesFor - votesAgainst)/(votesFor + votesAgainst)
tally = (tally + 1) / 2{normalize reputation}
return tally $\in [0, 1]$
```

Algorithm 5: Tally Annotation Reputation

```
calculateAnnotationReputationSpecificTrustBased (real votesFor, real votesAgainst, real initRep)
tally = (votesFor - votesAgainst)/(votesFor + votesAgainst)
tally = (tally + 1) / 2{normalize reputation}
weight = (minimum($N_{min}$, numberOfVotes))/$N_{min}$
rep = (1-weight)*initRep + weight*tally
return rep $\in [0, 1]$
```

Algorithm 6: Trust-Based Annotation Reputation

```
calculateAnnotationReputationSpecificCauchy (real votesFor, real votesAgainst, real initRep)
$scale = 1/\pi * arctan((\text{votesFor} - \text{votesAgainst} + \text{initRep})/\gamma) + 0.5$
return scale $\in [0, 1]$
```

Algorithm 7: Cauchy Annotation Reputation

```
calculateAnnotationReputation (Annotation a)
if no votes have been attached then
  reputation = a.initRep
else
  for each vote on annotation do
    if vote.for then
      votesFor += 1
    else
      votesAgainst += 1
    end if
  end for
  tally = (votesFor - votesAgainst)/(votesFor + votesAgainst)
  reputation = (tally + 1) / 2{normalize reputation}
end if
return reputation $\in [0, 1]$
```

Algorithm 8: General Annotation Reputation

4.2.2 Attaching Annotations to Learning Objects

The process we described so far requires annotations to be left by peers. We are assuming that a student is invited to leave annotations while he is experiencing a learning object (see Chapter 6). Algorithm 9 shows how annotations become attached to their learning objects. The annotation's initial reputation is set to be that of the annotator.

```
Inputs: Student, Learning Object, Annotation
student.reputation = calculateStudentReputation(student)
annotation.initRep = student.reputation
learningObject.attach(annotation)
Return: nothing
```

Algorithm 9: Make New Annotation

4.2.3 Which Annotations to Show to a Student

We can now clarify the procedure for determining which annotations to show to a student, including a detailed explanation of our proposed calculation for reputation.

We assume that there is currently an object in focus from the repository, for this user. For example, in the context of intelligent tutoring this object may have been selected according to its potential for enhancing the learning of this particular student, based on a modeling of that student's level of achievement and progress in the topic material to date. Within a general web-based information system, the object may simply have been selected by the user for possible consideration.

Determining which of the full set of annotations left on the object should be shown to the user is inspired by the work of Zhang et al. [92] which models trustworthiness based on a combination of private and public knowledge (with the latter determined on the basis of peers). Our process integrates i) a restriction on the maximum number of annotations shown per object ii) modeling the reputation of each annotation iii) using a threshold to set how valuable any annotation must be before it is shown iv) considering the similarity of the rating behaviour of users and v) showing the annotations with the highest predicted benefit. The overall procedure is displayed in Algorithm 10.

```
Arguments: learning object, student
{Global value $maxAnnotationsToDisplay$ determines the maximum number of annotations to display on a learning object}
{Global value $threshold$ determines the minimum reputation to allow an annotation to be seen by a student}
for each annotation attached to learning object do
  annotation.predictedBenefit = calculateAnnotationReputationSpecific(annotation, student){Algorithm 4}
end for
sort annotations in order of decreasing predictedBenefit
for first $maxAnnotationsToDisplay$ annotations do
  if annotation.predictedBenefit $>$ $threshold$ then
    show annotation
  end if
end for
Return: set of annotations to be shown
```

Algorithm 10: Show Annotation

4.2.3.1 Calculation of Reputability

Let A represent the unbounded set of all annotations attached to the object in focus. Let r^a_j = [-1, 0, 1] represent the jth rating that was left on annotation a (1 for thumbs up, -1 for thumbs down and 0 when not yet rated[2]). The matrix R has R^a representing the set of all ratings on a particular annotation, a, which also represents selecting a column from the matrix. To predict the benefit of an annotation for a user u we consider as Local information (using the terminology of Zhang [94], see Section 2.5) the set of ratings given by other users to the annotation. Let the similarity[3] between u and $rater$ be $S(u, rater)$. Global information (again using the terminology of Zhang [94], see Section 2.5) contains all users' opinions about the author of the annotation. Given a set of annotations $A_q = \{a_1, a_2, ..., a_n\}$ left by an annotator (author) q we first calculate the average interest level of

[2] Having an explicit representation of unrated annotations allows differentiation between rated and not yet rated, see Chapter 6 for an example.

[3] The function that we used to determine the similarity of two users in their rating behaviour examined annotations that both users had rated and scored the similarity based on how many ratings were the same (both thumbs up or both thumbs down). The overall similarity score ranged from -1 to 1.

an annotation a_i provided by the author, given the set of ratings R^{a_i} to the a_i, as follows:

$$V^{a_i} = \frac{\sum_{j=1}^{|R^{a_i}|} r_j^{a_i}}{|R^{a_i}|} \tag{4.1}$$

The reputation of the annotator q is then:

$$T_q = \frac{\sum_{i=1}^{|A_q|} V^{a_i}}{|A_q|} \tag{4.2}$$

which is used as the Global interest level of the annotation.

A combination of Global and Local[4] reputation leads to the predicted benefit of that annotation for the current user. We begin with a description of the primary metric we have used to date, a Cauchy cumulative distribution function (CDF) to integrate these two elements into a value from 0 to 1 (where higher values represent higher predicted benefit) as follows:

$$\text{pred-ben}[a, current] = \frac{1}{\pi}\arctan(\frac{(vF^a - vA^a) + T_q}{\gamma}) + \frac{1}{2} \tag{4.3}$$

where T_q is the initial reputation of the annotation (set to be the current reputation of the annotator q, whose reputation adjusts over time, as his annotations are liked or disliked by users); vF is the number of thumbs up ratings, vA is the number of thumbs down ratings, with each vote scaled according to the similarity of the rater with the current user, according to Eq. 4.4. γ is a factor[5] which, when set higher, makes the function less responsive to the vF, vA and T_q values.

$$v = v + (1 * S(current, rater)) \tag{4.4}$$

Annotations with the highest predicted benefit (reflecting the annotation's overall reputation) are shown (up to the maximum number of annotations to show, where each must have at least the threshold value of reputation).

4.2.3.2 Clarifying The Cauchy Formula

The formula which determines whether an annotation is shown or not is in Equation 4.3 is what we refer to as the Cauchy Approach. This has a number of attractive properties.

[4] Our usage of these terms is not perfectly analogous to Zhang's [94]; the differences are discussed in Section 8.5.5.

[5] A value of 0.2 was used for γ in our simulations.

Larger Number of Votes Given Greater Weight Consider 3 students as voting in favour of an annotation and 5 against it and contrast this with 30 students voting in favour and 50 voting against it. In each case, 37.5% of the student found the annotation useful. However, given that 10 times as many students have voted on the annotation in the second case, we want to take into account the greater certainty in the outcome given the larger number of voters. Contrast m votes for and n votes against and k(m) votes for and k(n) votes against for k > 1. Equation 4.3 is a linear scaling of

$$arctan(votes_{for} - votes_{against}) \tag{4.5}$$

Given that arctan is a strictly increasing monotonic function,

$$arctan(k(m-n)) > arctan(m-n) \tag{4.6}$$

and our approach will give a greater weight to larger groups of voters.

Probability Approaches But Doesn't Reach 0 and 1 Consider:

$$\lim_{x\to+\infty} f(x) = \frac{1}{2} + \lim_{x\to+\infty} \frac{arctan(x)}{\pi} = 1 \tag{4.7}$$

$$\lim_{x\to-\infty} f(x) = \frac{1}{2} + \lim_{x\to-\infty} \frac{arctan(x)}{\pi} = 0 \tag{4.8}$$

Given that arctan is a strictly increasing monotonic function, we know that the probability that an annotation will be shown or not approaches, but never reaches, 0 and 1. This is worthwhile as the usefulness of an annotation may be discovered at a later date, and it is then given a chance to be promoted. Conversely, if a well-regarded annotation is shown to be vacuous, the community has a chance to immediately begin decreasing its prominence. Consider the example of a clarifying annotation where a student made the connection, in a video that explains parameters, that procedure is another name for function. After this annotation was given high ratings, a new article was added which explained functions and clearly presented the various terms such as routine, function, procedure or method. After having read this article, students begin finding the previously useful annotation redundant and it begins to receive negative ratings from current students. The system is immediately responsive to this and each negative vote decreases the probability that this (once highly-regarded) annotation is shown to current students.

4.2.3.3 Clarifying The Tally Formula

Given an annotation a_i with a set of ratings from previous students, R, we calculate the predicted benefit for the student *current* as below for a Tally computation.

$$\text{pred-ben}[a_i, current] = \frac{\sum_{j=1}^{|R^{a_i}|} r_j^{a_i}}{|R^{a_i}|} \tag{4.9}$$

This formula gives us a value in the range of -1 to 1. When reasoning about reputation, we use a range of 0 to 1. Therefore, the value calculated here needs to be normalized (i.e. by adding 1 and dividing by 2, as in Algorithm 8).

4.2.3.4 Clarifying The Trust-Based Formula

We contrast the Cauchy technique, described above in Equation 4.3, with the Tally which is simply the average interest level as calculated in Equation 4.1 and the Trust-Based approach, which initially recommends annotations based on the reputation of the annotator (Equation 4.2) but increasingly bases its recommendation on the average interest level as more votes are registered on the annotation (in this work we uniformly distributed the weight of each recommendation based between completely the annotator's reputation when no votes were registered through to being completely based on the average interest level once a minimum number of votes were registered).

For the Trust-Based computation, given a set of annotations $A_q = \{a_1, a_2, ..., a_n\}$ left by an annotator (author) q we calculate the average interest level for the student *current* of an annotation a_i provided by the author, given the set of ratings R^{a_i} to the a_i, as follows:

$$\text{pred-ben}[a_i, current] = min(1, \frac{|R^{a_i}|}{N_{min}})\frac{\sum_{j=1}^{|R^{a_i}|} r_j^{a_i}}{|R^{a_i}|} + max(0, (1 - \frac{|R^{a_i}|}{N_{min}}))\frac{\sum_{i=1}^{|A_q|} V^{a_i}}{|A_q|} \tag{4.10}$$

The term $\frac{\sum_{j=1}^{|R^{a_i}|} r_j^{a_i}}{|R^{a_i}|}$ represents the community's rating of the reputability of the annotation. The term $\frac{\sum_{i=1}^{|A_q|} V^{a_i}}{|A_q|}$ represents the initial reputation of the annotation, which is set to be the annotator's reputation at the time the annotation was created. Over time, we want to place a greater weight on the community's view of the reputability of the annotation instead of the inherent reputations of the annotator. With each vote made on the annotation, we move the weight to a greater emphasis on the community's view of the reputability, until we reach a point where the community's perspective is the entire reputation and the reputation of the author is no longer considered. N_{min} is this point where we no longer consider the author's reputation. In our simulations we set[6] N_{min} to be 10.

4.3 Experimental Setup

In order to verify the value of our proposed model, we design a simulation of student learning. This is achieved by modeling each student in terms of knowledge levels (their understanding of different concepts in the course of study) where each learning object has a target level of knowledge and an impact that increases when the student's knowledge level is closer to the target (described in detail in Section 2.6.4.2). We construct algorithms to deliver learning objects to students in order to maximize the mean average knowledge of the entire group of students (i.e. over all students, the highest average knowledge level of each student, considering the different kinds of knowledge that arise within the domain of application).

As mentioned, one concern is to avoid annotations which may detract from student learning. As will be seen in Figures 4.2, 4.3 and 4.4, in environments where many poor quality annotations may be left, if annotations are simply randomly selected, the knowledge levels achieved by students, overall, will decline. This is demonstrated in our experiments by comparing against a Greedy God approach which operates with perfect knowledge of student learning gains after an annotation is shown, to then step back to select appropriate annotations for a student. The y-axis in our graphs shows the mean, over all students, of the average knowledge level attained by a student (so, averaged over the different knowledges being modeled in the domain).

As well as generating a random set of target levels for each learning object, we also generated a random length of completion (ranging from 30 to 480 minutes) so that we are sensitive to the total time required for instruction. The x-axis in each graph maps how student learning adjusts, over time. We used 20 students, 100 learning objects and 20 iterations, repeating the trials and averaging the results. For these experiments we ran the raw ecological approach, ecological with pilot and simulated annealing (as detailed in Section 3.4) for selecting the appropriate learning object for each new student; this has each student matched with the learning object best predicted to benefit her knowledge, based on the past benefits in learning achieved by students at a similar level of knowledge.

[6] Note, that we did not have the calibration of likelihood of an annotation being useful based on proportion and size of votes in order to apply the Chernoff Bound Theorem to compute this value.

Ratings left by students were simulated by having each student exposed to an annotation providing a score of -1 or 1; we simulated this on the basis of "perfect knowledge": when the annotation increased the student learning a rating of 1 was left, otherwise a -1 was left[7].

The standard set-up for all the experiments described below used a maximum of 3 for the number of annotations attached to a learning object that might be shown to a student; a threshold of 0.4 for the minimum reputability of an annotation before it will be shown; a value of 0.5 as the initial reputation of each student; and a value of 20% for the probability that a student will elect to leave an annotation on a learning object. An annotation can adjust the target level of instruction of a learning object by -5% to 5% and adjust the impact by -1 to 1. While learning objects are created by expert educators, annotations created by peers may serve to undermine student learning and thus poor annotation need to be identified and avoided.

4.4 Results

4.4.1 Quality of Annotations

We performed experiments where the quality of annotations from the group of simulated students varied. For each student we randomly assigned an "authorship" characteristic which provided a probability that they would leave a good annotation (defined as an annotation whose average impact was greater than 0). A student with an authorship of 10% would leave good annotations 10% of the time and bad annotations 90% of the time, while a student with an authorship of 75% would leave good annotations $\frac{3}{4}$ of the time and bad annotations $\frac{1}{4}$ of the time. In each condition, we defined a maximum authorship for the students and authorships were randomly assigned, evenly distributed between 0.0 and the maximum authorship. Maximum authorships of 1.0 (the baseline), 0.75, 0.25 and 0 were used.

The graphs in Figures 4.2, 4.3 and 4.4 indicate that our other approaches for selecting annotations to show to students (referred to as the Tally and Trust-Based in addition to the Cauchy) each performs well in a scenario with uniform distributions of student authorship. In environments where bad annotations outweigh good annotations, the 25%

[7] This perfect knowledge was obtained by running the simulated learning twice, once with the annotation and learning object, and once with just the learning object. A simulated student gave a positive rating if it learned more with the annotation and a negative rating if it learned more without.

and 0% authorship, we see that the Cauchy approach outweighs the two alternatives, and in the 0% environment the Tally approach dramatically under-performs.

These difference make sense when we consider the information the various approaches are based upon. The Tally approach uses exclusively the ratings that students have left for that annotation. Each annotation is judged on its own merits with no preconceptions. Unsurprisingly, this approach will perform poorly in environments dominated by poor annotations. The Trust-Based approach, in contrast, initially bases its recommendations on the annotation author's reputation, but as more votes are left on the annotation, it shifts its decision increasingly towards the ratings that have been left. This outperforms the Tally approach as it allows the identification of poor authors. The Cauchy approach always bases its recommendations on both the accumulated ratings from students and the reputation of the annotation author, which turns out to be a worthwhile approach in environments with many low quality annotations.

4.4.2 Cutoff Threshold

One approach to removing annotations or annotators from the system is to define a minimum reputation level, below which the annotation is no longer shown to students (or new annotations by an annotator are no longer accepted). A trade-off exists: if the threshold is set too low, bad annotations can be shown to students, if the threshold is set too high, good annotations can be stigmatized.

In order to determine an appropriate level in the context of a simulation, we examined cut-off thresholds for annotations of 0.5, 0.4, 0.3, 0.2, 0.1 and 0.0.

The results in Figures 4.5, 4.6 and 4.7 indicate that our algorithm is still able to propose annotations that result in strong learning gains (avoiding the bad annotations that cause the random assignment to operate less favourably) with the varying cut-off thresholds. The 0.0 threshold (indicative of not having a cut-off since this is the lowest possible reputation) underperformed in the early stages of the curriculum, indicating that having a threshold is worthwhile. The other various thresholds did not display substantial differences, which suggests a low cut-off threshold, such as 0.1 should be considered to allow the greatest number of annotations to be used by the system.

4.4.3 Explore vs. Exploit

Even for the worst annotators, there is a chance that they will leave an occasional good comment (which should be promoted), or improve the quality of their commentary (in

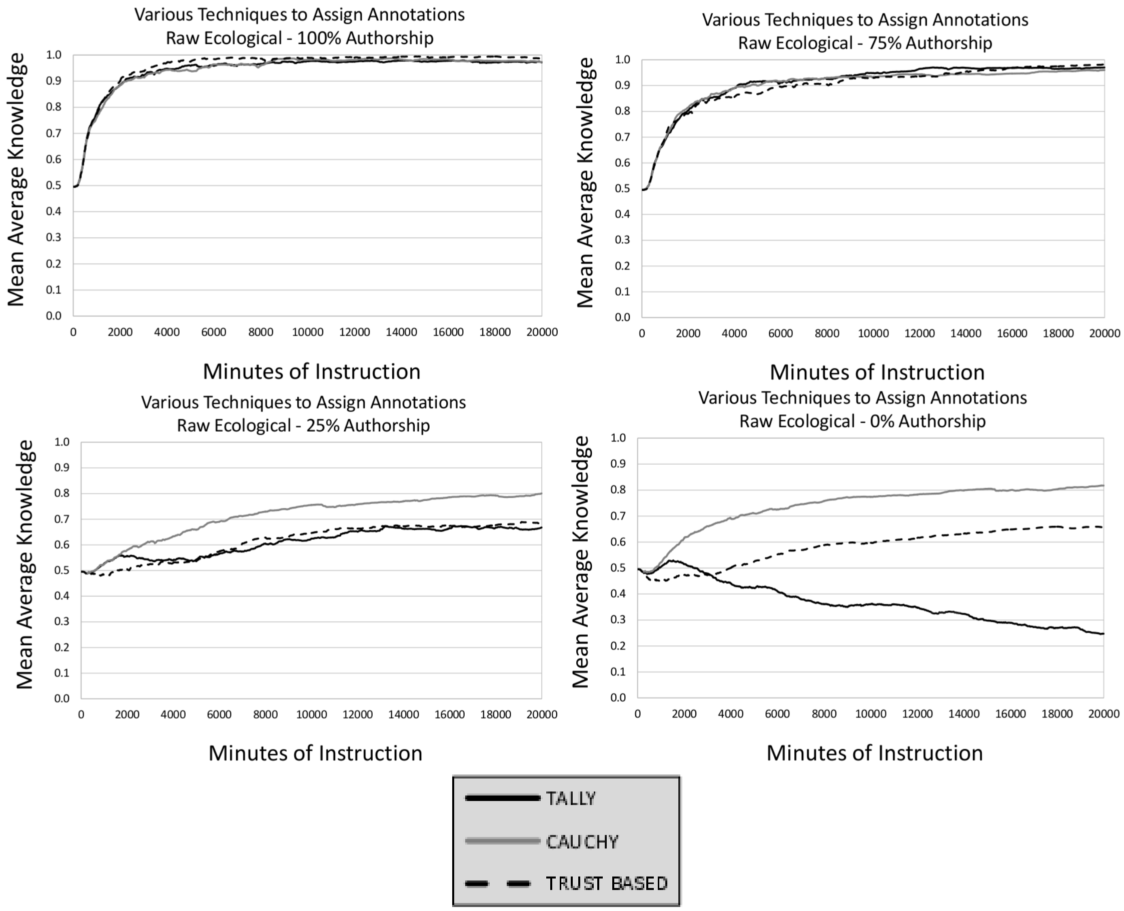


Figure 4.2: Comparison of Annotation Assignment Techniques - Raw Ecological

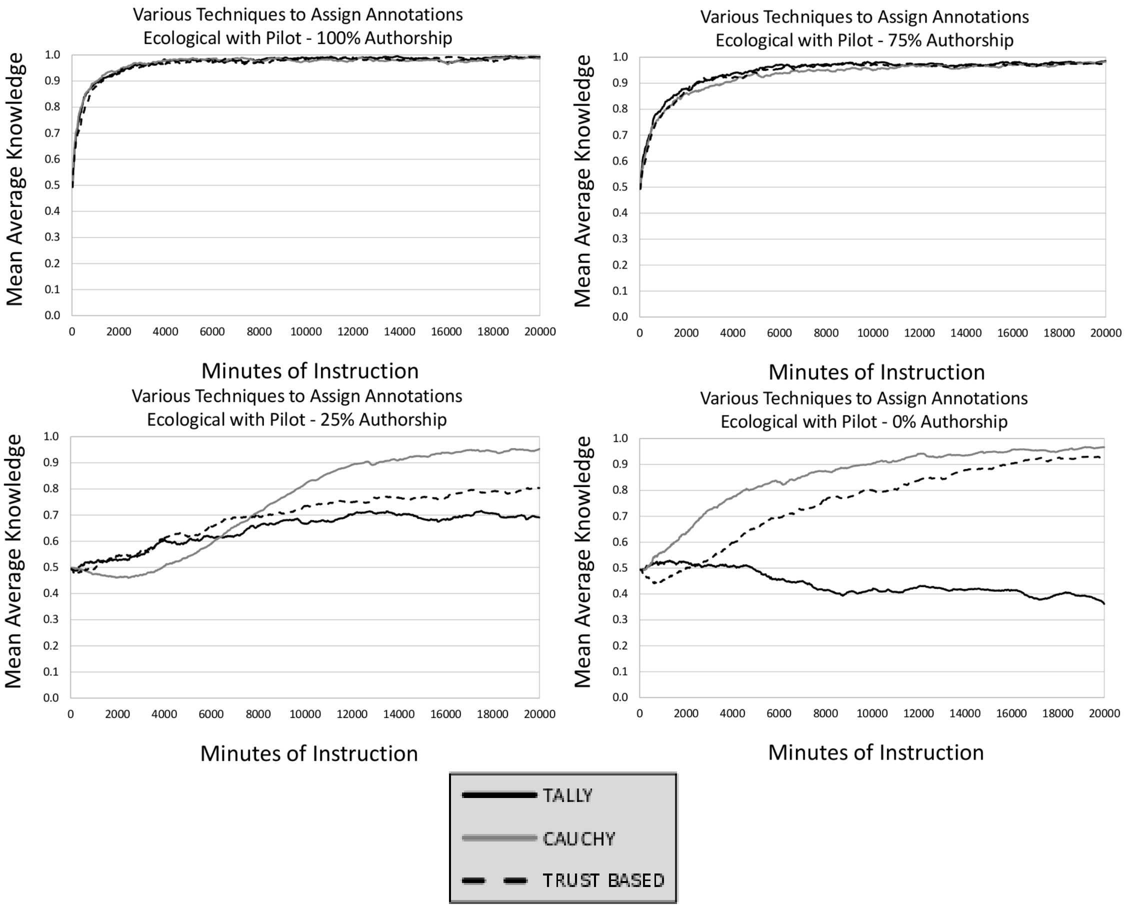


Figure 4.3: Comparison of Annotation Assignment Techniques - Ecological with Pilot

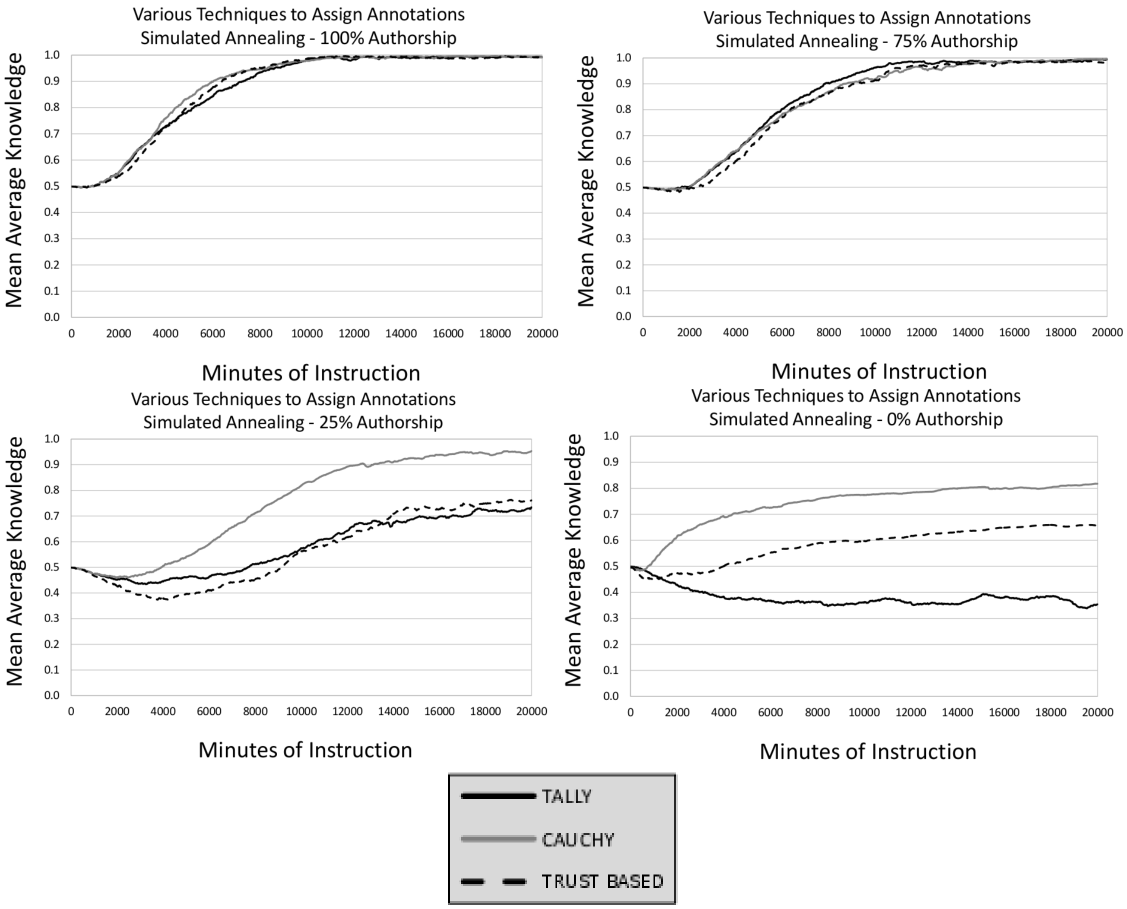


Figure 4.4: Comparison of Annotation Assignment Techniques - Simulated Annealing

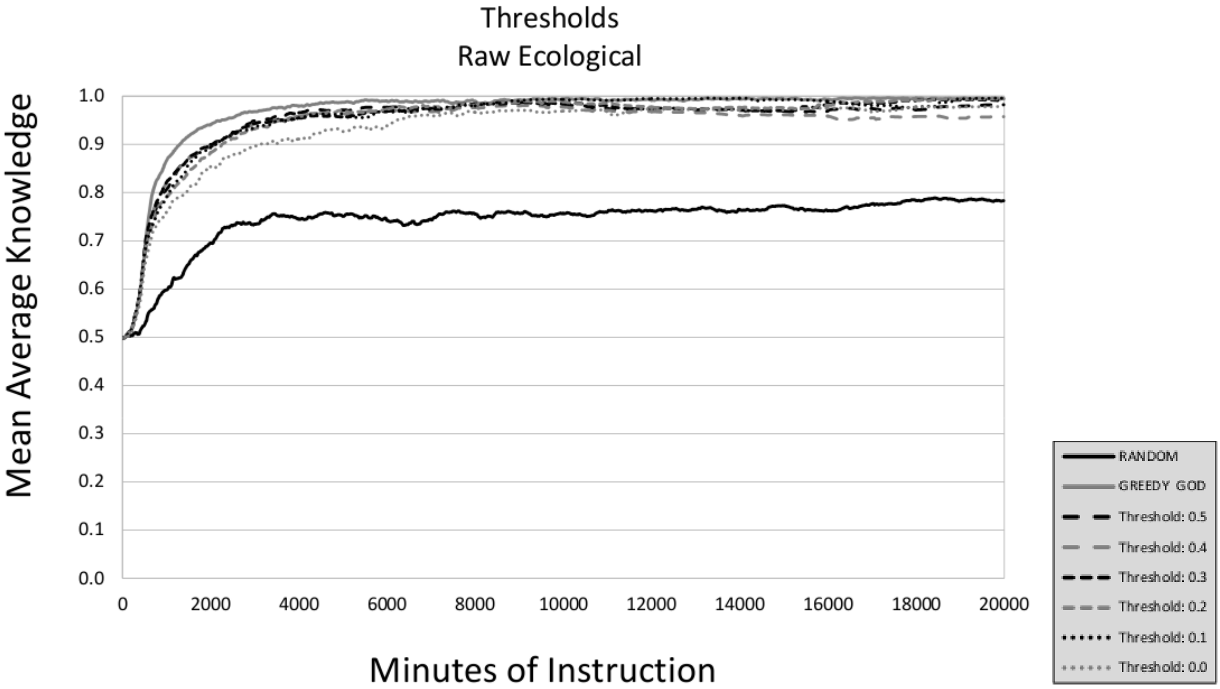


Figure 4.5: Comparison of Thresholds for Removing Annotations - Raw Ecological

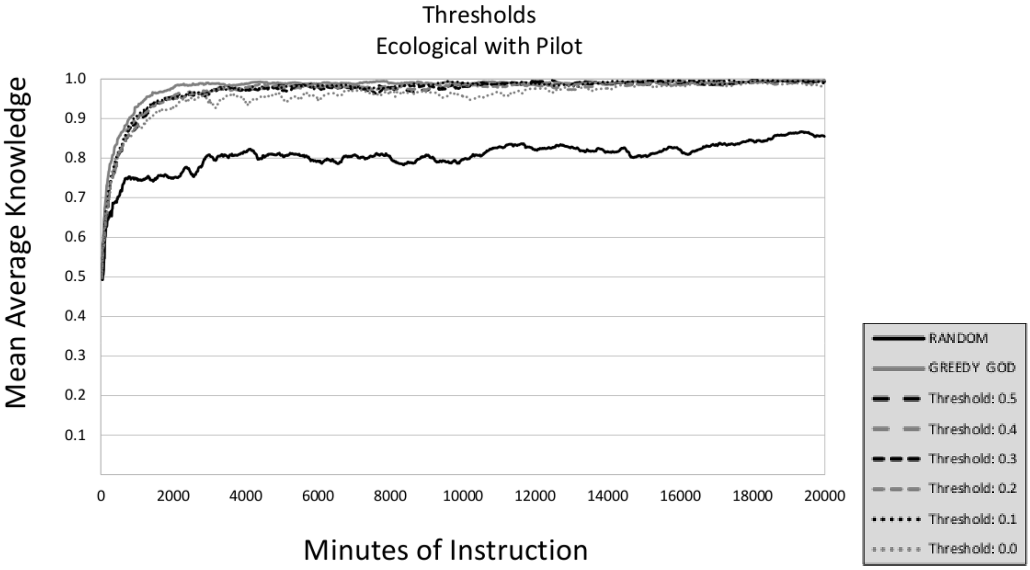


Figure 4.6: Comparison of Thresholds for Removing Annotations - Ecological with Pilot

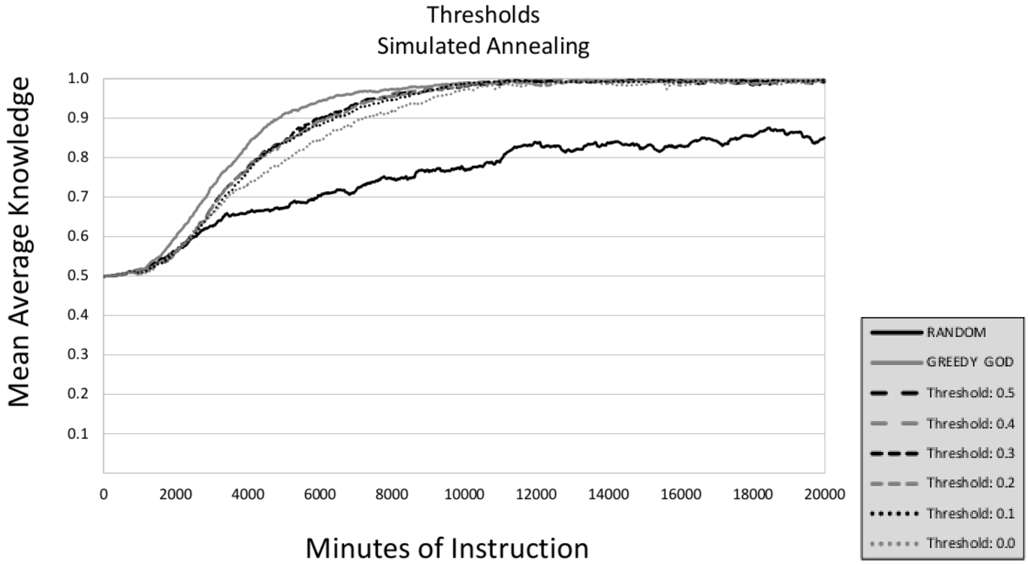


Figure 4.7: Comparison of Thresholds for Removing Annotations - Simulated Annealing

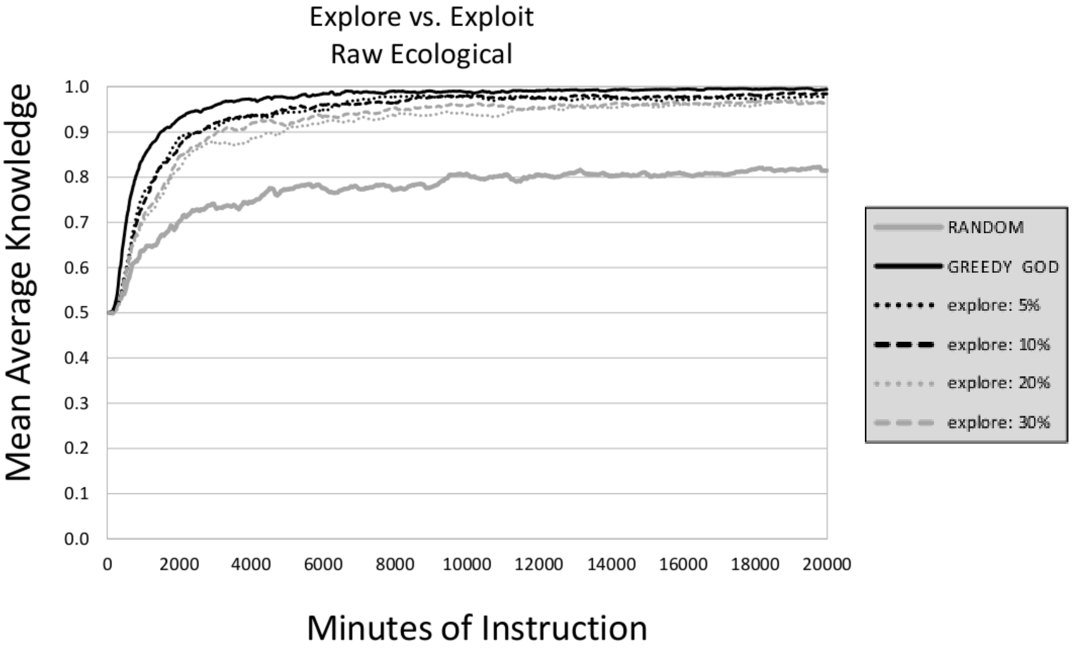


Figure 4.8: Explore vs Exploit: Raw Ecological

which case they should have a chance to be redeemed). For this experiment, we considered allowing an occasional, random display of annotations to the students in order to give poorly rated annotations and annotators a second chance and to enhance the exploration element of our work. We continued with the experimental setting of Section 4.3, where both local and global reputations of annotations were considered. We used two baselines (random and Greedy God) and considered 4 experimental approaches. The first used our approach as outlined above, the standard authorship of 100%, a cut-off threshold of 0.4 and a 5% chance of randomly assigning annotations. The second used an exploration value of 10%, which meant that we used our approach described above 90% of the time, and 10% of the time we randomly assigned up to 3 annotations from learning objects. We also considered conditions where annotations were randomly assigned 20% and 30% of the time.

Allowing a phase of exploration to accept annotations from students who had previously been considered as poor annotators turns out to still enable effective student learning gains, in all cases (see Figures 4.8, 4.9 and 4.10). Our algorithms are able to tolerate some random selection of annotations, to allow the case where annotators who would have

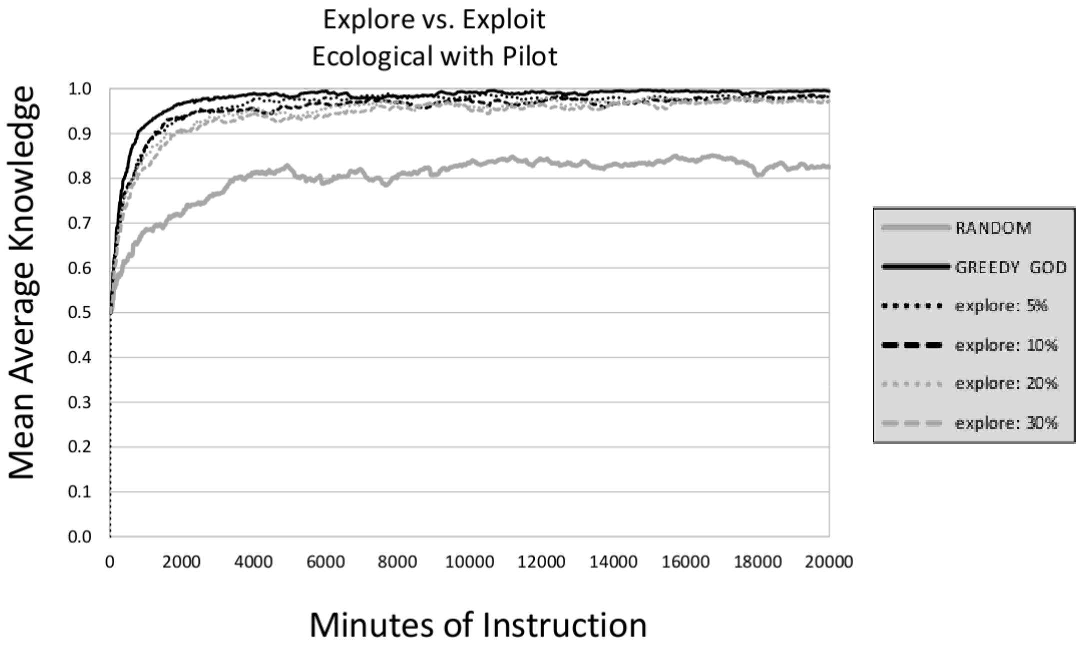


Figure 4.9: Explore vs Exploit: Ecological with Pilot

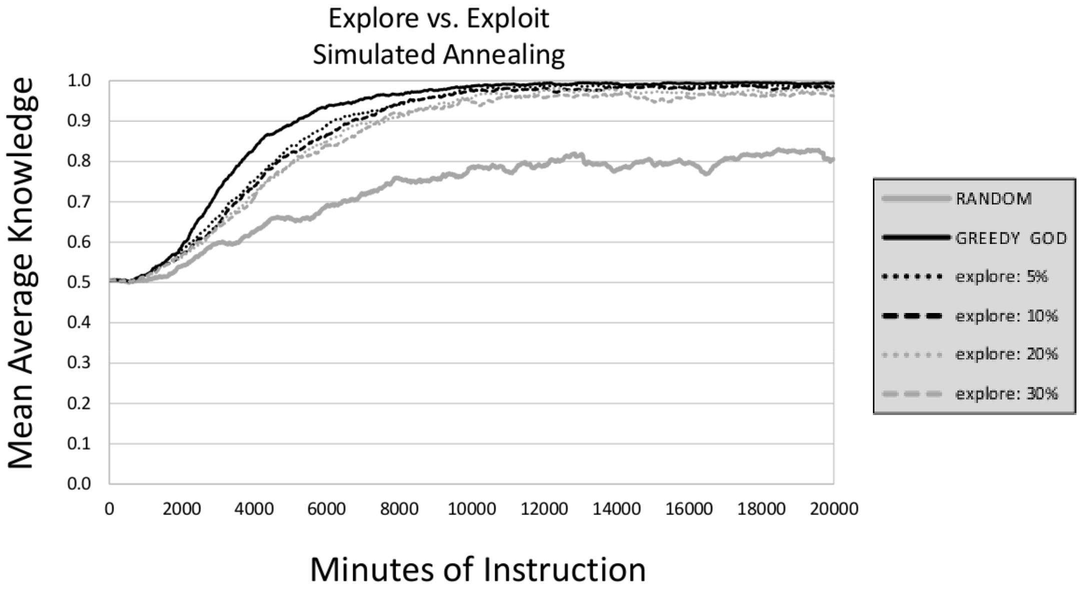


Figure 4.10: Explore vs Exploit: Simulated Annealing

otherwise been cut off from consideration have their annotations shared and thus their reputation possibly increased beyond the threshold (if they offer an annotation of value), allowing future annotations from these students to also be presented. In this simulation the authorship ability of students remained consistent throughout the experiment. Added randomness in the assignment of annotations would be more effective in situations where the authorship ability of students varied throughout the course of a curriculum. We discuss this option in Section 9.2.5.1.

4.4.4 Random Ratings

In order to test the simulation software for bugs, as described in Section 3.8 , in the source code where students rate whether or not an annotation was useful to them was replaced with a method that would randomly rate annotations. The expectation from this simulation was that each of the techniques for recommending annotations would degrade to provide results comparable with random assignment of annotations. For this simulation only a single iteration was run, which accounts for the greater volatility in the results Figure 4.11 shows the results with the three techniques operating normally, while Figure 4.12 shows the techniques with the ratings replaced with random ratings. These show the expected degradation to resemble the random assignment.

4.4.5 Scaling Votes

We were interested in examining the impact that our choice of scaling votes based on student similarity had on the recommended annotations. In our original simulation votes for and against a particular annotation were recorded then used as described above for the Tally, Cauchy and Trust-Based approaches to recommending annotations. When students had rated the same annotation in the past, the degree to which they agreed on their ratings was used to calculate a similarity score between each student. This score was used to influence the ratings previous students had left on annotations, strengthening the ratings left by similar students and weakening the ratings left by dissimilar students. Our approach takes advantage of pairs of students who consistently disagree on ratings to invert the ratings of these highly dissimilar peers.

In this experiment we now use a factor of 2 for the scaling provided by student similarity[8] and made the recommendations using the Cauchy approach. This means that a highly

[8] This falls out of the multiplication by sim used in Algorithm 4 (e.g. a vote For counts as 2 votes when student similarity is highest (1)).

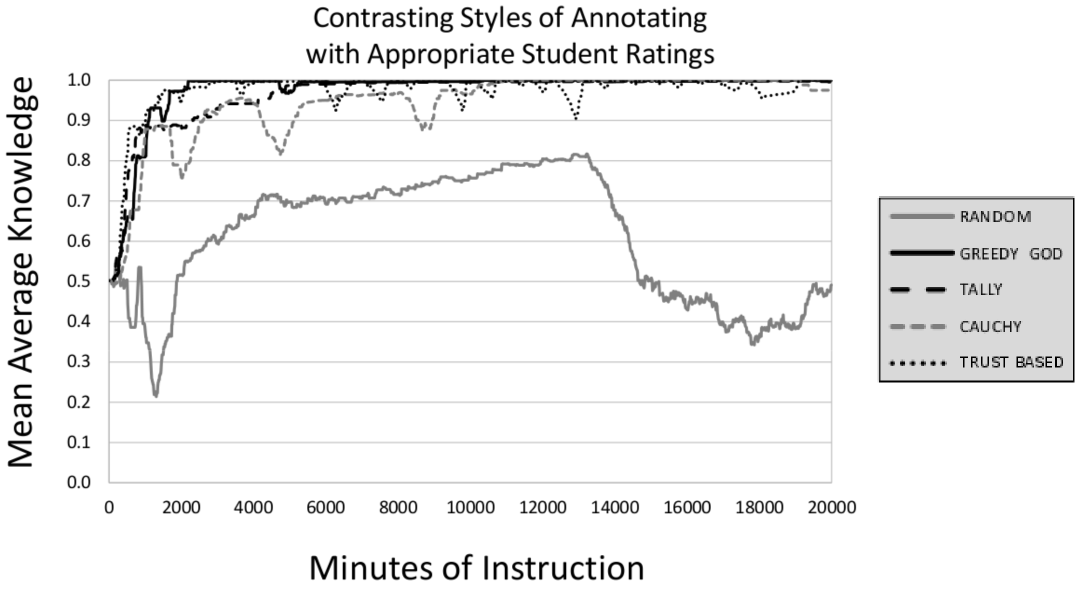


Figure 4.11: Comparison of Three Techniques for Recommending Annotations

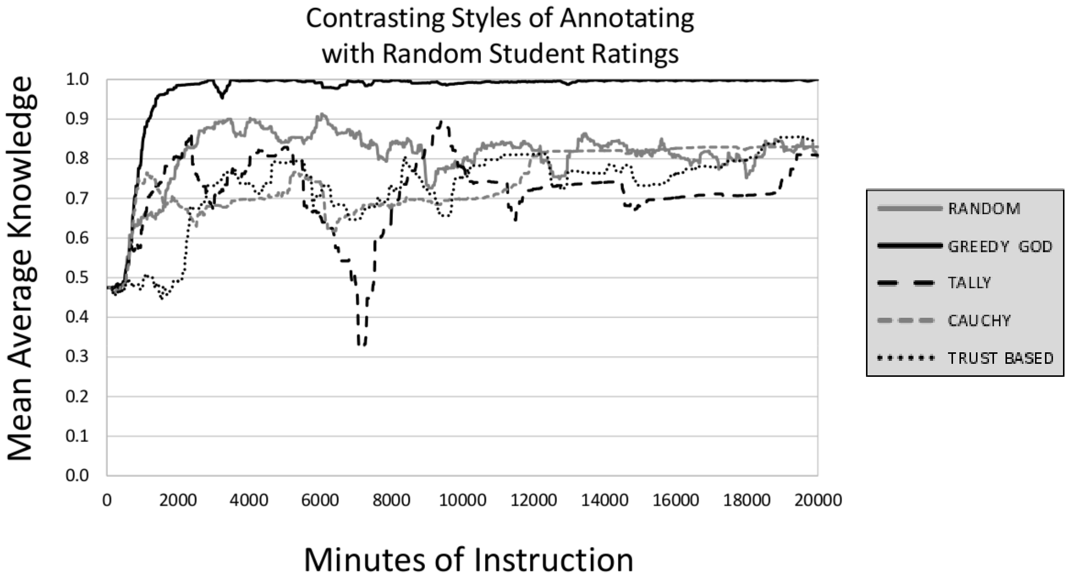


Figure 4.12: Three Techniques for Recommending Annotations with Student Ratings Replaced By Random Ratings

Student Similarity	Student Rating	Impact
1	Thumbs Up	2x Thumbs Up
0.5	Thumbs Down	1.5x Thumbs Down
0	Thumbs Up	Thumbs Up
-0.9	Thumbs Down	0.1x Thumbs Down

Table 4.1: Student Ratings Scaled by a Factor of 2

similar student's rating could be doubled (and would be considered equivalent to the ratings provided by two generic students). A student with a similarity of 0 would have their vote be treated normally. A consistently dissimilar student would have their rating weakened. For example, see Tables 4.1 and 4.2.

In an attempt to determine whether a factor of 2 would be a reasonable value, we ran our simulation and instead considered factors of 2, 5 and 10 when scaling ratings based on student similarity. For a factor a 5 or 10 this would give the possibility of highly similar students contributing a much larger impact. This would also have the result of inverting recommendations made by highly dissimilar students, recommending annotations that student had disliked and reducing the likelihood of seeing annotations that that student had liked. For example:

Student Similarity	Student Rating	Impact
1	Thumbs Up	10x Thumbs Up
0.5	Thumbs Down	5x Thumbs Down
0	Thumbs Up	Thumbs Up
-0.9	Thumbs Down	9x Thumbs Up

Table 4.2: Student Ratings Scaled by a Factor of 10

We ran the simulation for all three factors using the raw ecological approach to content sequencing (Figure 4.13), the pilot group approach (Figure 4.14), and simulated annealing (Figure 4.15).

The curves appear to be very closely matched to one another, but upon closer examination it can be seen that there are small variations, although in each case all three factors provide very similar results. At first this may be surprising, but upon reflecting these results make sense. For our simulation we attached the 3 most highly rated annotations to the learning object being seen by the student. The slight differences in results show that sometimes with different vote scaling factors this will result in different annotations being

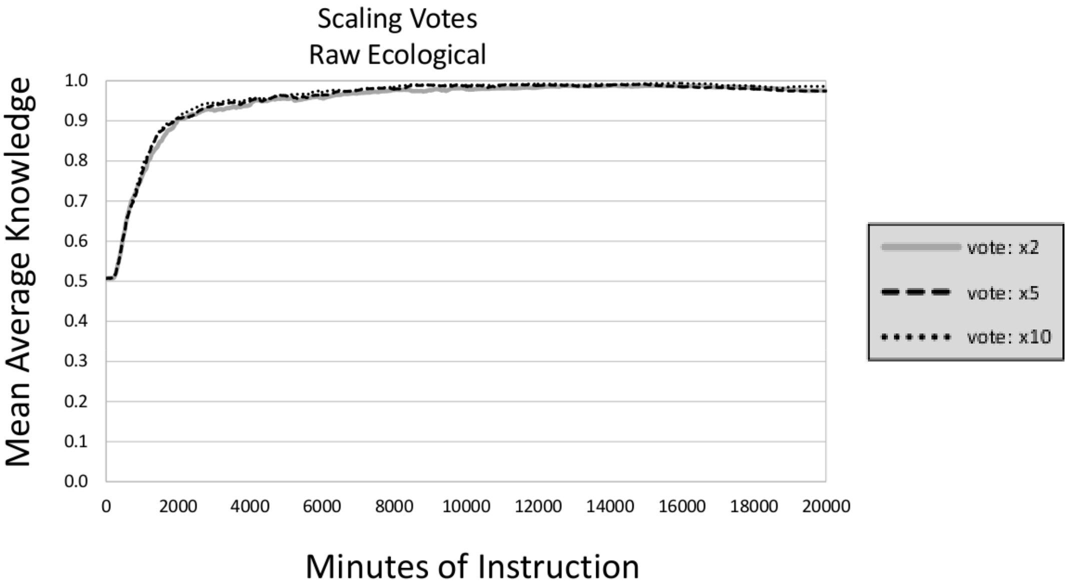


Figure 4.13: 3 Curves with Rating Scaling for Raw Ecological

assigned. It also shows that vote weighting does not result in a great deal of difference in the student experience, since each approach is making worthwhile (and quite similar) recommendations to the students. What is important, instead, is the proportion of ratings for and against an annotation and its ranking compared to other annotations, rather than the absolute value of the sum of votes.

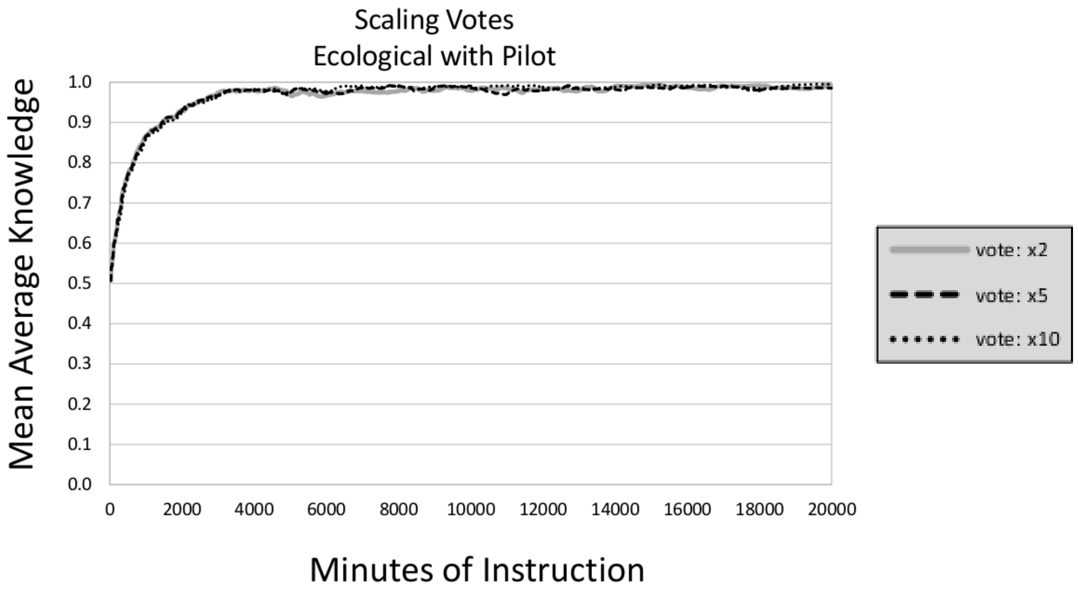


Figure 4.14: 3 Curves with Rating Scaling for Ecological with Pilot

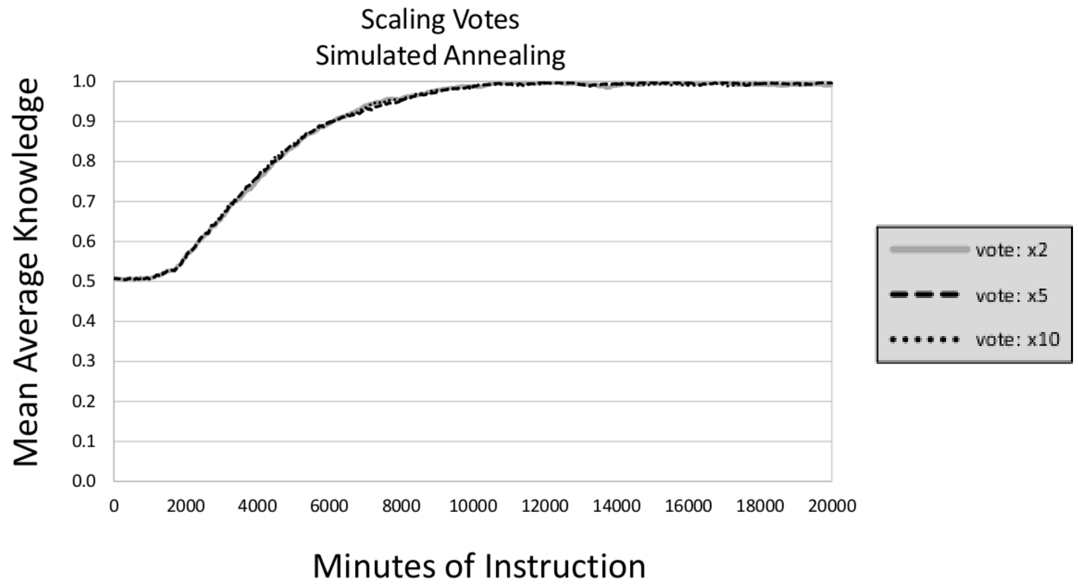


Figure 4.15: 3 Curves with Rating Scaling for Simulated Annealing

Chapter 5

Corpus

In this chapter, we outline our proposed technique for allowing peers to augment the corpus with divided learning objects. Our work takes as a starting point the curriculum sequencing described in Chapter 3 for the design of peer-based intelligent tutoring systems and first of all introduces an algorithm for selecting appropriate content (learning objects) to present to a student, based on previous learning experiences of peers. From here, our primary focus is on the process of enabling peers to augment the corpus of learning objects, proposing subdivisions of existing objects as valuable for guiding the learning of subsequent students. We then provide an algorithm that reasons about which learning objects are best to offer each new student, with this new repository that includes both the original objects and the new, subdivided entries.

The approach is entirely personalized: which objects are best for each student is based upon the student's current level of knowledge and the student's similarity with those peers who have either experienced the learning objects in the corpus or who have proposed the subdivision.

We demonstrate the value of our proposed framework through simulations of student populations experiencing learning objects, plotting the average knowledge level of all students when presented with objects determined by our algorithm, and in comparison with competing methods which do not allow the division of learning objects by students, or do not reason intelligently about assigning newly created learning objects to students based on the history of interactions by peers.

The capability of delivering personalized learning experiences for each student is emphasized by running a variety of conditions for the students, the peer-base and the overall repository. In particular, we model a class of students who have a preference for shorter learning objects, to show how our proposed algorithm is able to offer objects which provide the most effective increase in their knowledge levels. We also contrast: various approaches to curriculum sequencing and examine how they respond to the introduction of peer-divided learning objects; alternatives for differentiating between the assignment of parent (original) or children (divided) learning objects; groups of students with varying talent in creating valuable divisions of learning objects; simulations which allow or disallow personalization[1] and authorship[2], to show that we are able to offer the most effective student learning. Demonstrating the effectiveness of our proposed algorithm is achieved by showing the mean average knowledge attained by all students in simulations, in comparison with a Greedy God algorithm that has perfect knowledge for selecting the best objects and a method which randomly selects them instead.

Our approach to maintaining a corpus of learning objects is to provide students with tools to (optionally) divide non-atomic lessons[3], turning a single learning object into multiple learning objects. For example, if a student were assigned a book as a learning object, after reviewing it they might feel that only 3 chapters were worthwhile for what they were trying to learn. Extracting these from the book as a whole allows subsequent students to benefit from the original student's experience and quickly targets the part of the original learning object that was most worthwhile.

Since as each student experiences a learning object new objects may be created, these need to become new learning objects in the repository. The initial interaction history for each new learning object is assigned the interaction of the original learning object. The motivation for doing this is to enable the new learning object to be as likely as the parent to be recommended to new students; this serves to prime the new learning object for its place in the repository[4]. The newly created learning objects are then available to be assigned to students using the ITS.

Below are three examples which illustrate the need for and value of allowing corpus

[1] This allows information about the knowledge level of the student who did the division to be represented and then possibly influence what may be shown to a new student.

[2] This models a student's ability to identify the important concepts in a learning object. Students with a high authorship are more likely to create a high-quality learning object, while students with a low authorship are more likely to create a low-quality learning object.

[3] An atomic learning object would be something like a simulation that requires a programmer to refactor the material. A non-atomic learning object would include such things as textbooks, articles or videos that could be easily divided by students, given the right software tools.

[4] The introduction of a new learning object part way through the course of instruction creates a cold start problem. Inheriting the interaction history is, necessarily, an imperfect approach, as the entire point of creating a new learning object is for it to be different from the original in some way. However, the perspective was taken that this inherited history would be better than no history.

division by peers.

Example: Carol has been watching a supplemental video about Scheme for her CS 115 class, and found it to be not very useful except for one part that gave a very clear analogy for recursion she found useful. Within the ITS (at the completion of the lesson), she uses the clipping functionality to designate the beginning of this useful section and the end. A learning object is added to the system, the section she found useful.

Example: Bob has watched the section of the lecture that Carol highlighted above and found it useful. Because it is similar to the original learning object, the system next recommends the complete lecture to him. Bob finds it interesting as a whole, but doesn't like the last part of the lecture. He creates another new learning object with the beginning of the lecture and ignores the middle section that he saw in the previous interaction. This new learning object is added to the system.

Example: After a number of other students have watched the videos Carol and Bob created, the system has found that students independently found the first part of the video and the highlighted section Carol created more worthwhile than the original object. Advanced students found just the middle section useful, while less advanced students found the introduction and middle section useful. The last section wasn't worthwhile to anyone. The original learning object is stigmatized to the point that the system declares it not worthy of student attention and is ejected from the system. The middle section is recommended to advanced students and the beginning and middle are recommended to less advanced students.

5.1 Corpus Algorithms

We took the perspective that repetition was valuable and allowed students to interact repeatedly with the same learning object, each time modeling the interaction based on the student's current knowledge (see discussion in Section 9.2.2.3).

The algorithm that creates a new learning object, based on student selection of the most worthwhile parts, operates as follows. Once the student has selected a portion, a new learning object is created using only those parts. As mentioned, all data about previous interactions between students and the parent learning object are copied to the newly created learning object. Once new learning objects are introduced, the corpus grows to include divided objects as possible candidates for the *bestObject* selected for a student as in the Collaborative Learning Algorithm (included here as Algorithm 12, for ease of reading). The procedure to assign a learning object to a student is now as outlined in Algorithm 11. This Algorithm also clarifies when a student is given the opportunity to create a new divided learning object which is then added to the repository. When subsequent students are assigned learning objects they will be from the expanded repository that also includes the newly divided objects.

The main loop of Algorithm 11 tracks each time unit of instruction and ends when the course of study is completed. Its inner loop decides for each current student whether or not the student is available (i.e. is not currently working with a learning object) and therefore needs to have a learning object assigned. Once a student has completed interacting with a learning object, a post-test assessment is done and the interaction history of the object is updated. At this point, the student is invited to create a learning object[5]. The effects of this division are provided in Algorithm 13.

As mentioned above, we choose to allow repeated viewing of learning objects by students. This is consistent with the theories of psychologists such as Mace [51], Spitzer [76] and Pimsleur [62] and is reinforced by the “Blue’s Clues” case of school children viewing repeated lessons mentioned in Gladwell’s “The Tipping Point” [26]. Extensions to this work could allow each learning object to only be experienced once by a particular student. This would result in more complex reasoning about showing divided learning objects and their parents. This path is discussed further in Section 9.2.2.3.

[5] In a fully automated system, a suitable interface would then need to be designed to enable this to be done. See Section 9.2.6.1 for our discussion of this as future work.

```
Input: Repository of learning objects, set of students
for each time unit of instruction do
  for each student do
    if student is available then
      if not student's first learning object assignment then
        do post-test assessment of student
        attach interaction history of student to learning object
        update similarities between students{based on new assessment}
        {allow student to divide the learning object}
        if student creates a new learning object then
          generate learning object based on student and original learning object
          {This is done with Algorithm 13}
          add new learning object to repository of available learning objects
        end if
      end if
      assign student to a learning object, L{using CLA}
    end if
  end for
end for
Return: Repository of learning objects{includes old repository plus new objects}
```

Algorithm 11: Assigning Learning Objects in Expanded Corpus

```
Input the current-student-assessment
for each learning object: do
    Initialize currentBenefit to zero
    Initialize sumOfBenefits to zero
    Input all previous interactions between students and this learning object
    for each previous interaction on learning object: do
        similarity = calculateSimilarity(current-student-assessment,
        interaction-initial-assessment)
        benefit = calculateBenefit(interaction-initial-assessment,
        interaction-final-assessment)
        sumOfBenefits = sumOfBenefits + similarity * benefit
    end for
    currentBenefit = sumOfBenefits / numberOfPreviousInteraction
    if bestObject.benefit < currentBenefit then bestObject = currentObject
end for
if bestObject.benefit < 0 then bestObject = randomObject
```

Algorithm 12: Collaborative Learning Algorithm

```
Input: Learning object $lo$, student $s$
{After student $s$ completes interaction with learning object $lo$ and decides to suggest a
new learning object}
newLO = highlightWorthwhileSection($s$, $lo$)
for each $lo$.interaction do
    newLO.interaction[i] = $lo$.interaction[i]
end for
Return: newly created learning object, newLO
```

Algorithm 13: Divide Learning Object

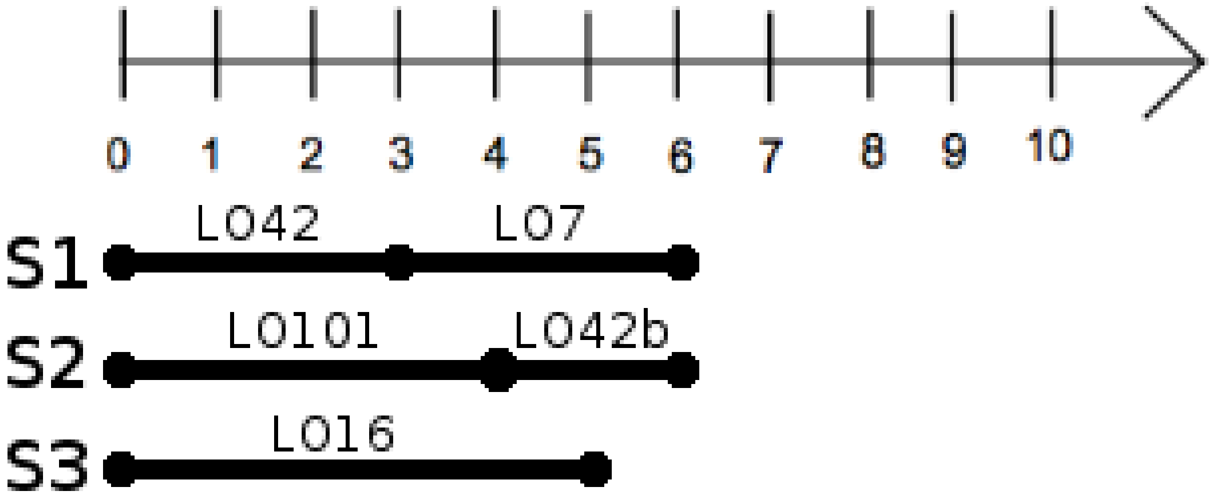


Figure 5.1: A Timeline of the Addition of Learning Objects

5.2 Clarifying the Assignment of Learning Objects in the Expanded Corpus

In this section we step through a detailed example at a high level in order to clarify how Algorithm 11 determines which learning objects are shown and when students have the opportunity to suggest divisions.

Consider Figure 5.1 as an example of three students (S1, S2 and S3) being assigned learning objects. This pictorial representation also clarifies the outer two loops of the algorithm. At time 0 (line 2 in Algorithm 11) the students all begin the course of study and have not been assigned a learning object yet. Each of the students, in turn (line 3), is evaluated to see if they are available (line 4), which they are, since the course of study has just started. The first learning object is being assigned to each student, so the Boolean condition in line 5 is false. In line 16 student S1 is assigned to Learning Object LO42, S2 is assigned to LO101 and S3 is assigned to LO16.

The next time unit is 1 (line 2 again). Each student is evaluated to see if they are available (line 4) and none of them are. They are all still interacting with the first learning object that was assigned to them. After this is time unit 2 (line 2) and each student is again determined to be unavailable (line 4). At time unit 3, S1 is determined to now be available, but S2 and S3 still are not.

The condition in line 5 now evaluates to true that this is not the first learning object for S1, so the subsequent code is executed. A post-test evaluation is made (line 6), which will be used both as the post-test assessment for the first learning object S1 experienced and his pre-test assessment for the second learning object he will experience. His initial pre-test from when he first entered the system and this post-test are attached to learning object 42 (line 7). His similarity to other students is re-calculated using his new assessment (line 8). S1 is given the option to create a new learning object (line 10), which he chooses to do, selecting 2 time units worth of material from the 3 time unit long original learning object. Learning Object LO42b is created (line 11, with precise details given in Algorithm 13) and this is added to the repository of possible learning objects to be assigned (line 13). S1 is now assigned to learning object LO7.

For time unit 4, S1 and S3 are not available; however, S2 is. The same sequence as previously performed by S1 is followed, except that S2 chooses not to create a new learning object. When it is time for a new learning object to be assigned (line 12), S2 is assigned the most beneficial learning object and in this example that turns out to be LO42b, the learning object S1 created.

5.3 Simulation

In order to simulate the learning achieved by students, we used the same approach as detailed in Section 3.3 and 3.7.

In our results, the x-axis represents the units of time that have passed so far for the instruction and the y-axis is the mean knowledge of the class as a whole, ranging from 0 (complete ignorance) to 1 (complete mastery), where an individual student's knowledge is calculated as the average over all $k \in K$.[6]

Note that when a peer-generated division creates a new learning object, the divided object begins with an interaction history inherited from its (longer) parent; the predicted benefit of the parent object is thus also carried over (and there is no adjustment in the calculation due to the divided object's shorter length). Once new students begin to interact with the divided object, the benefit calculation includes division by the length of object. Our description of the process used in the simulation is clarified through a series of lower-level examples which clarify the calculations that are done (with 1 dimension of knowledge used for simplification).

[6] In this simulation we used 6 k values; see Section 2.6.4.1.

Example Suppose we have multiple learning objects, LO, each with an identifier, a length of time, and multiple interactions with [index, [StudentID, initial assessment], [final assessment]] and students, S1, S2 and S3.

LO[1; 156 min; S1(B,A+)]
LO[2; 47 min; S3(A,A), S1(C,A-)]
LO[3; 210 min; S2(B-,A)]
LO[4; 23 min; S3(C,A-), S1(B,B)]
LO[5; 188 min; S3(C+,B)]

After these interactions, a new version of LO[1] is created, which is 50 minutes long. It inherits the interaction with S1 from its parent.

If a recommendation is being made for an A- student S2, and LO[1] and LO[1'] are both being considered, LO[1] would have a predicted benefit of $\frac{1}{1+2} \times 4 \div 156 = 0.0085$. LO[1'] inherits the interaction history of its parent and begins with the same predicted benefit (0.0085).

Suppose that LO[1'] were shown to S2 and the student had a positive learning experience moving from a pre-test of A- to a post-test of A+. Now if S3 (assessed as an A) were presented with a choice of LO[1] or LO[1'], this time LO[1'] would have a higher predicted benefit, calculated to be $\frac{1}{2} \times (((\frac{1}{1+3} \times 4) \div 156) + ((\frac{1}{1+1} \times 2) \div 50)) = 0.0132$.

On the other hand, if S2 had had a negative experience with LO[1'] moving from a pre-test of A- to a post-test of B then the choice for S3 would turn out to be LO[1], as the predicted benefit of LO[1'] would reduce to be $\frac{1}{2} \times (((\frac{1}{1+3} \times 4) \div 156) + ((\frac{1}{1+1} \times -2) \div 50)) = -0.0068$.

In our simulation (detailed in Algorithm 14) a new lesson is created 20% of the time when a student interacts with a learning object, representing a group of students who are highly motivated to contribute to the system. After division, the length of the newly created learning object is set to half of the original learning object[7]. The impact of the newly created learning object was set to be proportionate to the new length, halved in this case[8]. This impact was then adjusted to be slightly higher or lower, depending on the skill of the student who proposed the division. The impact for each knowledge dimension, k, is increased or decreased by 10%. The likelihood of an increase (rather than a decrease) was set to be the *authorship* rating of the student who made the division. Student authorship ratings were randomly generated in the range [0,1] and can be thought of as the student's ability to discern higher and lower quality parts of learning objects.

[7] This simplification, that division is always halved, may be relaxed in future work and extensions made for arbitrary division of objects; see Section 9.2.5.1.

[8] This was done in order to model the fact that only part of the entire possible benefit from the learning object was experienced.

```
Input: Learning object $lo$, student $s$
{After student $s$ completes interaction with learning object $lo$}
if $lo$.generation $< 3$ then {Allow a max of 3 divisions}
  if Random.prob(divide-value) $< 0.20$ then {20% chance a new division is proposed}
    newLO = lo
    newLO.generation = $lo$.generation + 1
    newLO.length = $lo$.length / 2 {division always halves an object}
    for each $lo$.impact do
      newLO.impact[i] = $lo$.impact[i] / 2 {inherits half the impact of the parent}
      if Random.prob(author-value) $<$ s.authorship then
        newLO.impact[i] = newLO.impact[i] * 1.1 {this models improved impact}
      else
        newLO.impact[i] = newLO.impact[i] / 1.1 {this models decreased impact}
      end if
    end for
    {achieve personalization}
    for each $lo$.targetLevel do
      diff = ($s$.knowledgeLevel[i] - $lo$.targetLevel[i]
      newLO.targetLevel[i] = lo.targetLevel[i] + $0.1 \times$ diff
    end for
    for each $lo$.interaction do
      newLO.interaction[i] = $lo$.interaction[i]
    end for
  end if
end if
Return: newly created learning object, newLO
```

Algorithm 14: Divide Learning Object Simulation

The target level of instruction (see Section 3.4) for each dimension of the learning object's knowledge was moved 10% closer to the current knowledge of the student making the division. We refer to this as *personalizing* the newly created learning object towards its author. This represents the idea that a student who is not as advanced as the target level of instruction of the learning object will tend to select the simpler ideas presented, while a student who is more advanced than the target level of instruction of the learning object will tend to select the advanced concepts.

Example Suppose we have a learning object LO[5; 188 min; S3(C+,B)].

Now suppose in a simulation S1 interacts with LO[5], has a good interaction (pre-test assessment C, post-test assessment B+) and creates a new, divided learning object, LO[5'] (line 4 in Algorithm 14).

The lesson time of LO[5'] will be set to be half that of LO[5], 94 minutes in this case. Suppose the impact of LO[5] was 0.42, initially the impact of LO[5'] would be set to be 0.21 (half the original impact). This value is then scaled based on the authorship ability of the student (their ability to select the better elements from a learning object). For instance, a student with authorship modeled as good would end up causing an adjustment to the impact of LO[5'] to 0.231 (per line 10 of Algorithm 14).

Suppose that the target level of instruction of LO[5'] is 0.602 (inheriting initially the target level from LO[5]). We now consider the knowledge level of S1. Suppose that this is 0.575. Then the target level of instruction for LO[5'] would be adjusted per lines 17 and 18 to be: $0.1 \times (0.602 - 0.575) = 0.0027$ closer to the student's knowledge, or 0.5993. In essence, this learning object would now be somewhat better suited to more advanced students.

For each experiment, identical learning objects and students were used for the various approaches. After the simulation, each student and learning object was reverted to its original state for the next run. This allows us to remove the variability from randomly differing groups of students and libraries of learning objects when comparing the effects of the various conditions. The interaction history of the original, undivided learning object was copied to the newly created learning object. After the division, each is tracked and considered independently. Ultimately, this algorithm recommends which learning object, including newly created learning objects, should be assigned to a particular student at a particular point in her course of study using Algorithm 11.

For our first baseline experiment without divisions, we used a larger set of students (see Figure 3.4). For the remaining experiments 20 students interacted with 100 learning objects over 20 iterations and the averaged results are presented[9]. Students were simulated as having certain levels of knowledge as a result of their post-test assessments. To be realistic, we modelled some possible error with the assessment. This was achieved by considering a Gaussian distribution with a mean of 0 and a standard deviation of 0.1, yielding a number that would adjust the student's assessment value to be either somewhat higher or lower (see Section 3.6). Variable time length learning objects were used. A total time of instruction of 20,000 time units was used and each learning object ranged from 30 time units to 480 time units[10]. The impact of each learning object ranged from -0.05:0.05 for a 30 time unit lesson, scaled proportionately by length of lesson[11] through to an impact of -0.8:0.8 for a 480 time unit lesson.

In order to plot learning curves, the average knowledge ($\in [0, 1]$) of all students is plotted against their progress in the course of study. Algorithms perform well when the average knowledge attained by students is high. A set of algorithms (the same employed in Chapter 3 and 4) to select learning objects for students were run, to demonstrate the value of the proposed approach. **Random Association** associates each student with a randomly assigned learning object; **Greedy God** chooses the best possible interaction for each student for each trial[12]. These two curves are the benchmarks (low performance and "the ideal"). Three variations of Algorithm 1 were then run. **Raw Ecological** has each student matched with the learning object best predicted to benefit her knowledge; **Pilot Group** has a subset of the students (10%) assigned, as a pilot group, systematically to learning objects - these interactions are used to reason about the best sequence for the remaining 90% of the students; **Simulated Annealing** provides a good approximation of the global maxima for a large search space. During the first 1/2 of the trials there is an inverse chance, based on the progress of the trials, that each student would be randomly associated with a lesson; otherwise, the ecological approach was applied. This approach provides a nice balance between exploration and exploitation.

[9] This was done to reduce random noise in the results.

[10] Time units can be considered roughly analogous to a minute. Due to the abstract nature of a simulated model these could easily be considered any arbitrary length unit. It is best for the reader not to get too locked into thinking of them as a specific, real world unit.

[11] For example, a 90 time unit lesson would have impact ranging from -0:15:0:15, three times a 30 time unit lesson.

[12] This is achieved by giving the algorithm full access to the fine-grained knowledge levels of the students and learning object, testing what the outcome would be for every possible interaction, then choosing the best interaction for each student for each trial.

5.3.1 Experiments

In this section we briefly outline the set of experiments we chose to conduct and the primary purpose of each. For these experiments we added noise to the pre- and post-test assessments. In Section 3.6 we confirmed that our model is robust to assessment error.

- **Divisions of Learning Objects** We first allow students to propose divisions of learning objects and add these divided learning objects to the repository. We then track mean average student knowledge. The expectation is that student learning will not be harmed by the addition of these peer-created learning objects.
- **Differentiation of Original and New Learning Objects** Since a divided learning object inherits the interaction history of its parent, the predicted benefit of this divided object is initially equal to the predicted benefit of the parent. This experiment is set up to explicitly favour either the parent or the divided object to be presented first to the student. The expectation is that regardless of whether the parent or child is favoured first, the overall student learning achieved will be comparable.
- **Personalization and Authorship** Whereas in all experiments to this point both personalization and authorship features are implemented, in this experiment we remove one or the other condition. The expectation is that student learning will be best when both conditions are in effect, reinforcing the value of our particular approach.
- **Varying Authorship Ability in Students** This series of experiments varies the probability that the students will provide good divisions of learning objects. The expectation is that student learning will be effective even in the presence of students who are less skilled in dividing learning objects, but that the best levels of student learning will be observed when the population of students is skilled in dividing.
- **Short Attention Span Students** This experiment demonstrates the personalization that is available with our approach. We simulate a set of students with short attention spans who should prefer to be shown shorter learning objects. The expectation is that the learning achieved by these students is still effective.

5.4 Results

5.4.1 Divisions of Learning Objects

For this experiment we contrasted (i) divided learning objects, assigned to students using Algorithm 11 with each of the ecological, pilot[13] or simulated annealing approaches with (ii) assigning the newly created learning object randomly or (iii) using the greedy god approach. In this experiment, students are able to propose divisions of learning objects and the repository of learning objects is extended to include divided learning objects as well as the original ones. As mentioned above, learning objects are divided 20% of the time when a student interacts with them, and at most a learning object can have three generations (that is, only itself and its children can be divided, its grandchildren can not). These results are displayed in Figure 5.2.

Personalization and authorship were used for this simulation to provide a richer modeling of students and learning objects in the simulation. In other words, the impact of the learning object is increased when the division is performed by a student who is well skilled in authorship, leading to an increased chance that the divided learning object would be shown to a new student if the authorship levels of the dividing student were higher; in addition, the target level of the learning objects is brought closer to that of the dividing student to increase the chances of showing this object to like-minded students.

5.4.1.1 Results

These results put the efficacy of reasoning about divided learning objects in context between the baselines of the random (lower end) and greedy god (ideal case) approaches. The steep curve of the greedy god condition is due to the large number of learning objects, 500, and the newly created learning objects, all of which this approach can immediately capitalize on.

We see that with the divided learning objects, the strengths of the various approaches still hold. The simulated annealing under-performs initially, but delivers the best result by the end of the experiment and the pilot does best early on, at the cost of a poor experience for the pilot group who prime the system for their classmates.

[13] During the pilot phase, when those students who prime the system initially experience learning objects, they may suggest divisions the same as any other students. These learning objects are added to the repository.

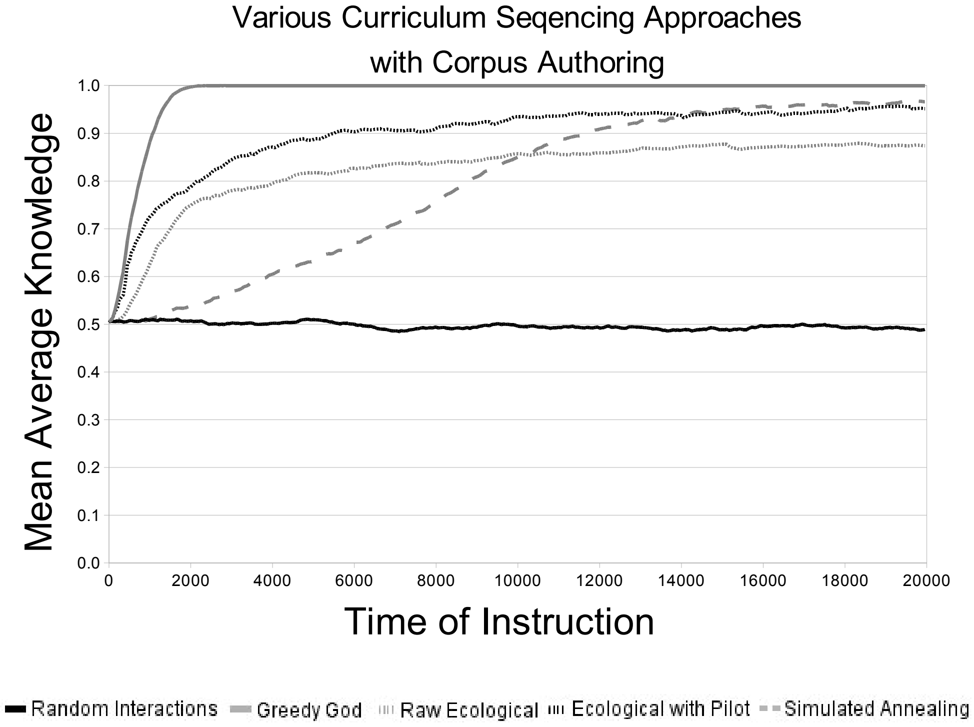


Figure 5.2: Divisions of Learning Objects

These results suggest that it is not harmful to allow peer-based authoring, even in situations with highly variable authorship quality.

5.4.2 Differentiation of Original and New Learning Objects

Previously (see Chapter 3) our simulations considered a fixed library of learning objects, all of which are present at the beginning. Divided learning objects, therefore, presented a new challenges as they introduce new learning objects part way through the simulated students' course of study. This is an example of the cold-start problem: how to recommend a new item with no interaction history.

In this work, we took the perspective that inheriting the history of interactions from the parent learning objects was the best approach to "prime" the newly created learning object. As the parent and child each has further interactions with students, they can be differentiated between and (potentially) each recommended to populations of student who would benefit from these interactions.

One challenge is that after creation and the attachment of these interactions, the parent and child learning objects will have identical interaction histories, and therefore will have equal predicted benefit for any particular student. Which to recommend in the case of this tie is then the question that presents itself. For this experiment, we contrasted yielding to the parent learning object versus yielding to the child learning object (which we refer to as Reverse). These results are displayed in Figure 5.3.

5.4.2.1 Results

The similar results from giving priority to the original learning object or the newly created learning object suggests that either approach allows for the divided learning object to be differentiated from the original, and that a systemic bias towards one or the other isn't a large concern.

How the algorithm breaks ties and differentiates between the original and newly created learning objects does not seem to compromise the performance of our algorithm. This confirms that we are effectively learning which objects to present to students in order to achieve the best benefit, regardless of the initial assignments of objects to students.

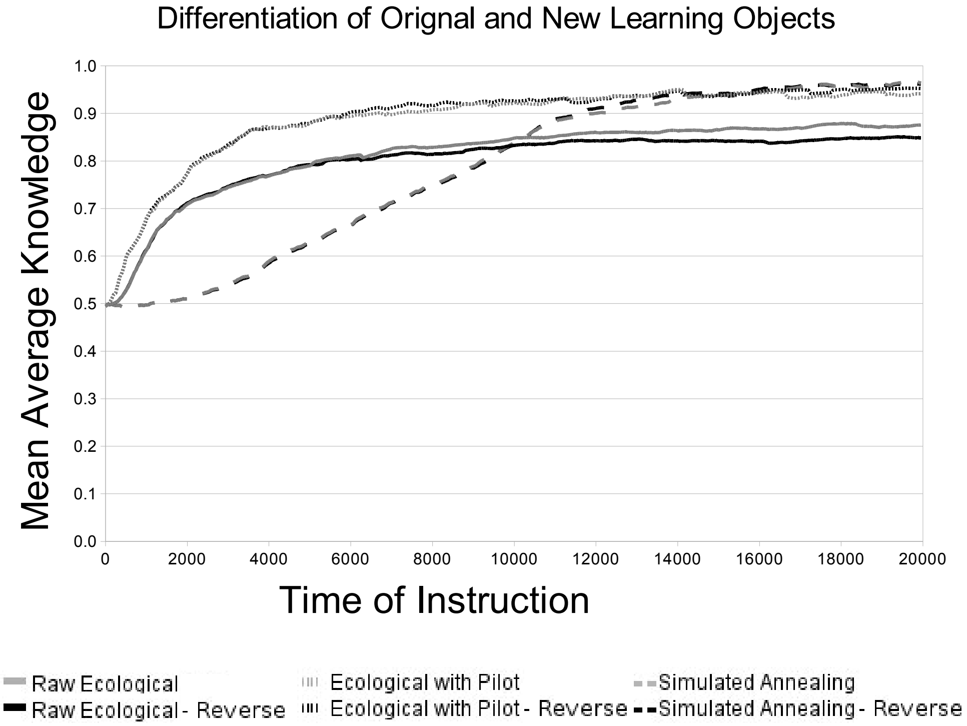


Figure 5.3: Differentiation of Original and New Learning Objects

5.4.3 Personalization and Authorship

These experiments examine the impact of personalization and authorship (as described above) on the division of learning objects. We examined 4 conditions in each of the curriculum sequencing approaches:

1. personalized and authored
2. personalized and not authored
3. not personalized and authored
4. neither personalized nor authored

The intention for this experiment was to determine if, in fact, our approach is capitalizing on better or worse authorship ability among the student population and a tailoring of created content to better suit students who are similar to the authoring student.

When authorship was not used, we set the impact of the newly created learning object to be exactly $\frac{1}{2}$ the original learning object's impact. This gives both learning objects the same educational benefit per minute of instruction (see lines 6 and 8 in Algorithm 14). When personalization was not used, the target level of instruction was the same in the newly created learning object as in the original. In addition, we emphasize the possible gains in personalization and authorship by adjusting the increasing factor used in Algorithm 14 (lines 10, 12 and 18) from 10% to 25%. These results are displayed in Figure 5.4.

5.4.3.1 Results

We see from this that our approach is capitalizing on student authorship ability and personalization. The curves incorporating both of these outperforms the curves incorporating one of them, which both in turn outperform the curves incorporating neither.

5.4.4 Varying Authorship Ability in Students

We contrasted different populations of quality of student authoring by allowing the student authorship characteristic to be all good (impact will always be raised), all bad (impact will always be lowered) and ok, i.e. evenly distributed between both (50% chance of being raised and 50% chance of being lowered). These results are shown in Figure 5.5.

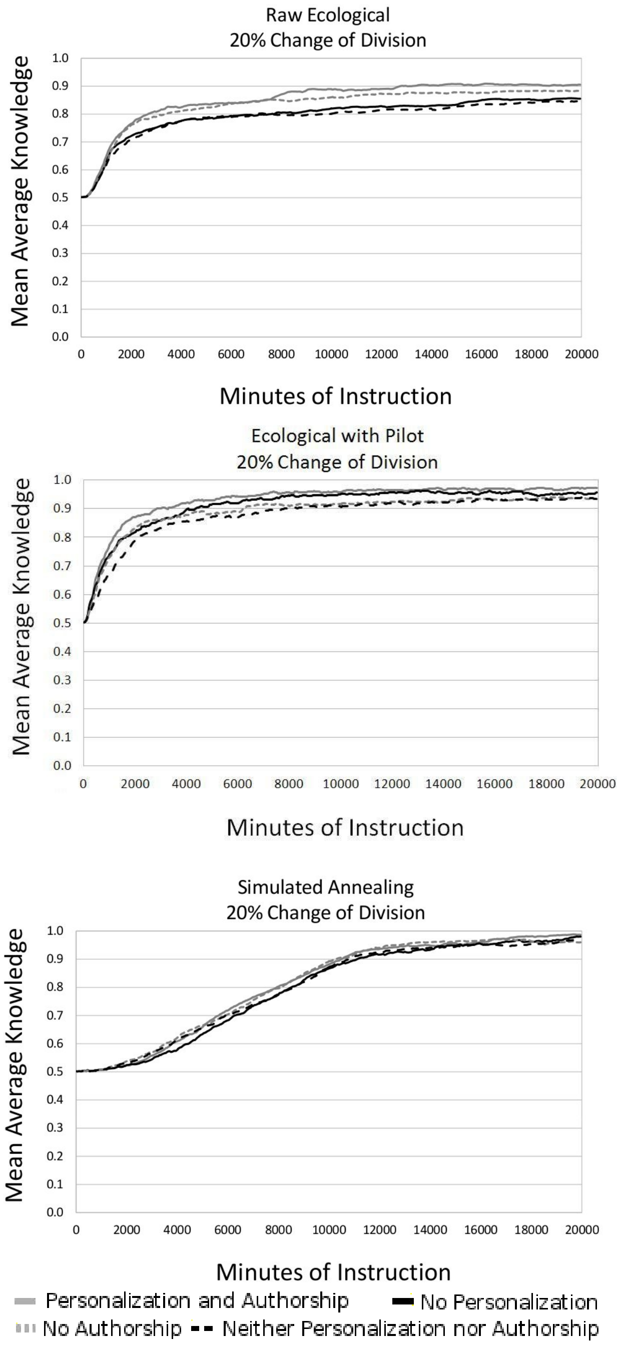


Figure 5.4: Personalization and Authorship

5.4.4.1 Results

These 3 results show that, for each of the curriculum sequencing approaches, our technique was able to make useful recommendations to students. The similar results in the raw ecological and the ecological with pilot for both the good and the ok authorship demonstrates the techniques are very good at finding worthwhile authorship (an even divide of good and bad authoring is comparable to all-good authoring). The minimal under-performance of the all-bad authoring results provides support for the perspective that even in the case of very low authorship quality, a reasonable curriculum (based primarily on the original learning objects) will be delivered.

5.4.5 Short Attention Span Students

We incorporated the idea of students with different learning styles by modeling students with short attention spans. We scaled the impact of learning objects for these students by increasing the shortest third (30 to 180 time units) of lessons by 25%, and decreasing the impact of the longest third (330 to 480 time units) of lessons by 25%. This was contrasted, for the raw ecological, ecological with pilot and simulated annealing curriculum sequencing approaches, with a group of students who were identical except for not having short attention spans (which we refer to as SAS). These results are shown in Figure 5.6.

5.4.5.1 Results

The results for each of the various curriculum sequencing approaches show that our approach handles student populations with different learning needs (in this case length of lesson), making recommendations such that appropriate learning objects are shown. With the raw ecological approach, the short attention span group underperformed slightly, while in the ecological with pilot approach it outperforms and in the simulated annealing they are evenly matched.

The Pilot group out-performance can be explained by the more extensive interaction history provided by the pilot group. This makes it easier for the system to identify the superior performance from shorter lessons, and therefore preferentially recommend these from the start. The raw ecological and simulated annealing must gather data before there is sufficient history to systematically assign shorter learning objects to students.

It is important to point out that our approach was not tailored to students benefiting from or being penalized for varying lesson lengths. This was incorporated into the simulation of the interactions between students and learning objects, but our approach reasoned

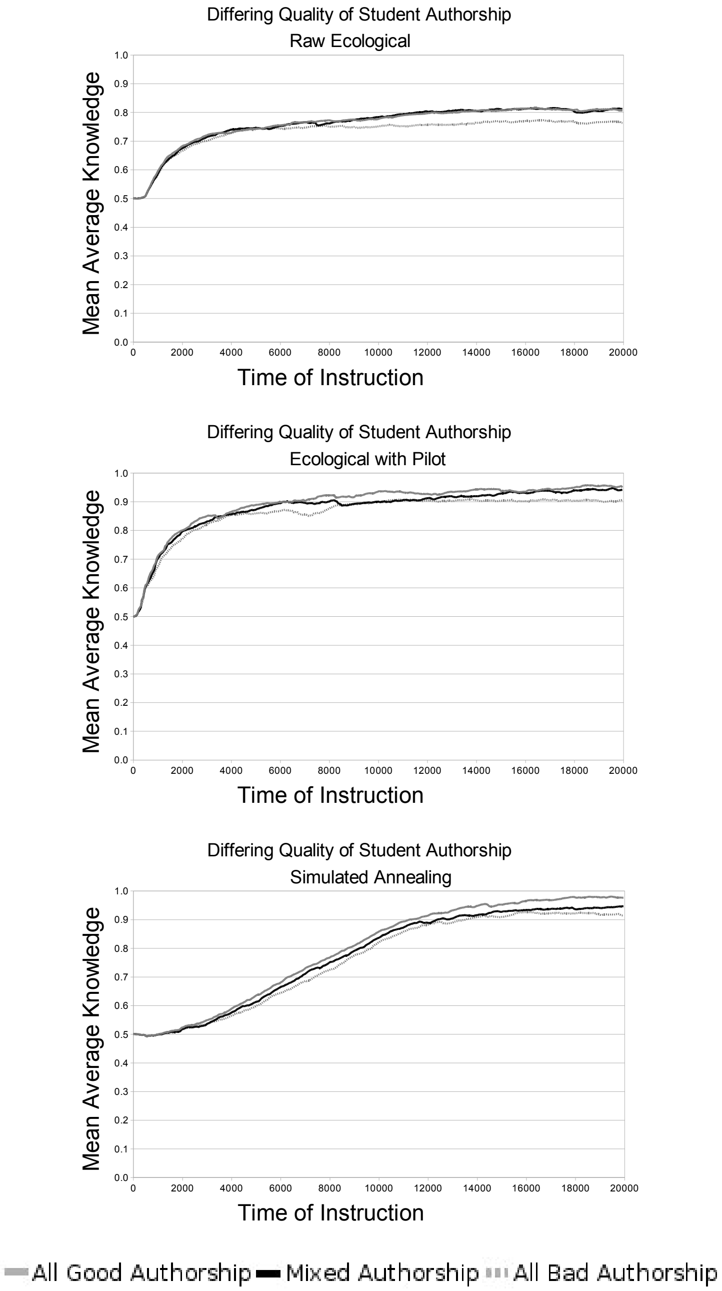


Figure 5.5: Varying Authorship Ability in Students

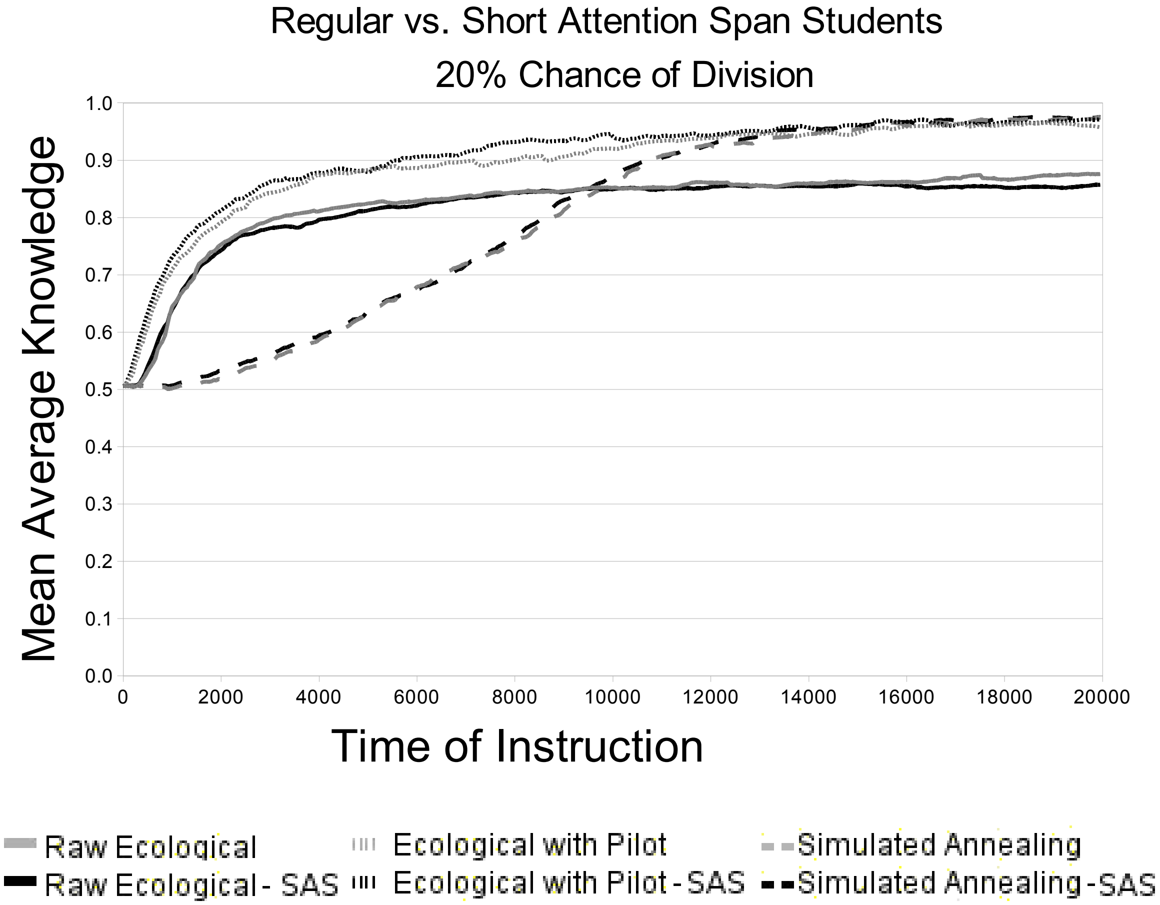


Figure 5.6: Short Attention Span Students

about interactions using only the pre- and post-assessments of each interaction. Using only this data our approach was able to tailor the curriculum to the special needs of this group.

Chapter 6

Driver

Putting the three techniques detailed in Chapters 3, 4 and 5 together allows the benefits provided by each to be utilized for student interactions. Once the curriculum sequencing selects a particular learning object, specific annotations can be attached to it to further personalize the interaction. As students use the corpus approach to subdivide lessons, a richer repository of learning objects is available to the curriculum sequencing to improve future recommendations. This chapter shows the overall algorithms used, illustrated with examples.

Algorithm 15 shows the full range of processes from the point of view of each student, through the overall course of study. Algorithm 25 clarifies the data structures used and the key procedures in place, including when assessments are done. We use this as the high level driver when presenting examples to illustrate the overall processing of our model, in this chapter. The Collaborative Learning Algorithm (CLA) is presented again for reference in Algorithm 16. Algorithms 17-23 are called by the high level driver (Algorithm 25) and are thus included in this chapter as well.

```
for each minute of instruction do
  {LOs each have a duration}
  for each student $s$ do
    if $s$ has no learning object then
      {new learning object}
      assign $s$ to a learning object, $L${using CLA}
      for each annotation assigned to learning object $L$ do
        {based on ratings from previous students}
        determine predicted benefit for $s$
      end for
      assign annotations attached to $L$ with highest predicted benefit to $s$
    end if
    if time == $L$.completionTime then
      finalize $s$ interaction with $L$
      for each annotation of $L$ shown to $s$ do
        $s$ assigns rating to annotation
      end for
      update annotation reputations
      update annotator (student) reputations
      update similarities between students{based on new ratings}
      if $s$ leaves a new annotation then
        generate annotation based on $s$ (annotator)
        attach new annotation to learning object interacted with
      end if
      if $s$ creates a new learning object then
        newLO = highlightWorthwhileSection($s$, $lo$)
        for each $lo$.interaction do
          newLO.interaction[i] = $L$.interaction[i]
        end for
        add newLO to repository of learning objects
      end if
    end if
  end for
end for
```

Algorithm 15: Driver Algorithm

```
Input the current-student-assessment (CSA)
for each learning object: do
    Initialize currentBenefit to zero
    Initialize sumOfBenefits to zero
    Input all previous interactions between students and this learning object
    for each previous interaction on learning object: do
        similarity = calculateSimilarity(CSA, interaction-initial-assessment(IIA))
        benefit = calculateBenefit(IIA, interaction-final-assessment)
        sumOfBenefits = sumOfBenefits + similarity * benefit
    end for
    currentBenefit = sumOfBenefits / numberOfPreviousInteraction
    if bestObject.benefit < currentBenefit then
        bestObject = currentObject
    end if
end for
if bestObject.benefit < 0 then
    bestObject = randomObject
end if
```

Algorithm 16: Collaborative Learning Algorithm

```
Arguments: Student
if numberOfAnnotationsCreatedByStudent == 0 then
    {If the student hasn't made any annotations,}
    {assign their reputation to 0.5}
    Reputation = 0.5
else
    Reputation = 0
    {Calculate the average reputation of all annotations made by the student}
    for each previous annotation left by student do
        reputation += annotation.reputation
    end for
    Reputation /= numberOfAnnotationCreatedByStudent
end if
Return: Reputation (0.0:1.0)
```

Algorithm 17: Calculate Student Reputation

```
Arguments: Student, Learning Object, Annotation
student.reputation = calculateStudentReputation(student){Algorithm 17}
annotation.reputation = student.reputation
learningObject.attach(annotation)
Return: nothing
```

Algorithm 18: Make Annotation

```
Arguments: Current Student, Annotation Student
votedTogether = 0
votedAgainst = 0
for each annotation both students voted on do
    if currentStudent.vote == annotationStudent.vote then
        votedTogether += 1
    else
        votedAgainst += 1
    end if
end for
similarity = (votedTogether  votedAgainst)
similarity /= (votedTogether + votedAgainst)
Return: Similarity [-1.0:1.0]
```

Algorithm 19: Similarity in Rating Annotations

```
calculateAnnotationReputationSpecific (Annotation a, User u)
if a has no votes then
    reputation = a.initRep{Return the reputation of the annotator at the time the annotation was created}
else
    for each vote on annotation do
        sim = similarity(u, voterUser){Algorithm 19}
        if vote.for then
            votesFor += 1 + 1 * sim
        else
            votesAgainst += 1 + 1 * sim
        end if
    end for
    if using tally for reputation then
        reputation = calculateAnnotationReputationSpecificTally(votesFor, votesAgainst, annotation.initRep){Algorithm 5} ‘
    else if using trust-based for reputation then
        reputation = calculateAnnotationReputationSpecificTrustBased(votesFor, votesAgainst, annotation.initRep){Algorithm 6}
    else
        {using Cauchy for reputation}
        reputation = calculateAnnotationReputationSpecificCauchy(votesFor, votesAgainst, annotation.initRep){Algorithm 7}
    end if
end if
return reputation $\in [0, 1]$
```

Algorithm 20: Annotation Reputation

```
Arguments: learning object, student
for each annotation attached to learning object do
    predictedBenefit = calculateAnnotationReputationSpecific(annotation, student){Algorithm 20}
end for
for top X annotations do
    if annotation.predictedBenefit > 0 then
        show annotation
    else
        show random annotation with annotation.reputation > threshold
    end if
end for
Return: nothing
```

Algorithm 21: Show Annotation

```
Input: Learning object lo, student s
{After student s completes interaction with learning object lo and decides to suggest new learning object}
newLO = highlightWorthwhileSection(s, lo)
{annotations in the highlighted section are retained}
for each lo.interaction do
    newLO.interaction[i] = lo.interaction[i]
end for
Return: newly created learning object, newLO
```

Algorithm 22: Divide Learning Object

```
Input: Assessment a1, Assessment a2
sim = 1 / (1 + difference(a1,a2)){Algorithm 24}
Return: sim
```

Algorithm 23: The Similarity Between Two Assessments

```
Input: Assessment a1, Assessment a2
distance = 0.0
for each dimension of knowledge, $k$ do
    distance += $|a1.k - a2.k|$
end for
Return: distance{This fills in the value for difference}
```

Algorithm 24: The Manhattan Distance Between Two Assessments

```
while studentsSeekingInstruction == true do
  addAnyNewStudents(){Any new students get added to a queue and introduced here}
  for i from 1 to students.totalNumber do
    for j from i to students.totalNumber do
      similarity[i,j] = calculateSimilarity(student[i],student[j]){Algorithm 23}
      similarity[j,i] = similarity[i,j]
    end for
  end for
  for each studentsSeekingInstruction do
    studentSeekingInstruction.remove(currentStudent)
    pretestAssessment = assessment(currentStudent)
    currentStudent.pretestAssessment = pretestAssessment
    lo = CollaborativeLearningAlgorithm(pretestAssessment){Algorithm 16}
    currentStudent.lo = lo
    annotations = showAnnotation(lo,currentStudent){Algorithm 21}
    assign(lo,annotations,currentStudent)
    {during interaction, currentStudent can makeAnnotation Algorithm 18}
    {during interaction, currentStudent can annotation.rate}
  end for
  for each studentFinishedInstruction do
    posttestAssessment = assessment(currentStudent)
    attach(currentStudent.lo,currentStudent.pretestAssessment,posttestAssessment)
    newLO = divideLearningObject(lo,currentStudent){Algorithm 22}
    if newLO is not null then
      learningObjectRepository.add(newLO){new LOs available for CLA}
    end if
    studentSeekingInstruction.add(currentStudent)
  end for
end while
```

Algorithm 25: High-Level Driver

6.1 Example: Recently Deployed System With No Previous Interactions

Given a new deployment, consider student Adam as the first student using it. He enters at line 2 of Algorithm 25. Initially his similarity to other students is calculated (line 5); however, since there are no other students who have used the system, nothing is done here. Adam is assessed (line 11) and this value is recorded as his pre-test assessment (line 12). Applying the Collaborative Learning Algorithm (CLA) to his pre-test assessment (line 13) the system finds a learning object, LO42, for Adam to experience.[1]. Annotations to be shown to Adam are determined (line 15); however, since he is the first student using the system there are no annotations to attach to LO42. Adam begins his interaction with the learning object (line 16). Algorithm 25 continues to loop while Adam interacts with the learning object. Each time it gets to line 20 it evaluates whether or not Adam has finished his interaction. During the interaction, Adam is given the opportunity to leave annotations attached to a specific part of the learning object (line 17 and Algorithm 18). If he were viewing annotations left by other students, he would also be given the opportunity to rate them (line 18).

6.2 Example: First Student Using the System Completes First Interaction

After Adam completes his interaction with LO42, line 20 evaluates to true and therefore lines 21 to 27 are evaluated. First, Adam is assessed again (Algorithm 25 line 21) and this is recorded as his post-test. His pre-test and post-test assessments are attached to LO42 (line 22) to be used when assigning learning objects to future students. Now that the interaction is complete, Adam is given the opportunity to create a new version of LO42 with the parts he felt were most important retained, and the redundant parts removed (shown in Algorithm 22). The system uses these highlighted sections provided by Adam to create a new learning object (line 23). Copies of all previous interactions (at this point only Adam's pre-test and post-test assessments) are copied and attached to the newly created

[1] In this case, there are no interaction histories for the CLA to use in its reasoning. Since all learning objects would have a predicted benefit of 0 without any data to reason about, a learning object is assigned randomly (line 17 in Algorithm 16). In the experiments we ran, initial data was generated for all learning objects using the raw ecological, ecological with pilot and simulated annealing approaches. This part of the algorithm is omitted for clarity.

learning object (Algorithm 22 line 5) and the newly created learning object, LO107, is added to the repository of objects that can be assigned to students (Algorithm 25 line 25).

6.3 Example: A Second Student Uses The System

A second student, Barbara, begins using the system. Like Adam, she enters at line 3 of Algorithm 25. Her similarity to all other students (only Adam at this point) is calculated in line 5, and since Barbara has no assessment yet, she and Adam are evaluated as being completely dissimilar. Barbara is assessed in line 11, and this is recorded as her pre-test assessment. The CLA is applied to her assessment to recommend a learning object to her (line 13). Because she was determined to be completely dissimilar to Adam, his experience with LO42 is discounted and not used in the determination. Again, a random learning object is assigned (line 17 in Algorithm 16), LO87 and no annotations are attached since no student has ever used this learning object before (Algorithm 25 line 15). Learning Object LO87 is assigned to Barbara and she begins her interaction with it (Algorithm 25 line 16). The algorithm continues to loop while Barbara interacts with the object (line 20 keeps evaluating to false) and Barbara is given the opportunity to leave annotations (line 17) which she does while she interacts with the object (Algorithm 18). If there were any existing annotations attached to this learning object (there are not), she would be given the opportunity to rate them (line 18).

6.4 Example: Second Student Continues Using System

After completing her interaction with the learning object, her post-test assessment is determined (Algorithm 25 line 21) and attached along with her pre-test assessment to LO87 (line 22). Barbara does not suggest a division (line 23). Her similarity to all other students (only Adam in this case) is calculated again (line 5), and now with assessments for both of them they are determined to have a similarity of 0.67 (the assessment is recorded for Adam as well in line 6). Similarity ranges from 0, maximally dissimilar to 1 maximally similar (which would imply the same assessment). Barbara is compared to Adam's assessment before he interacted with LO42. Barbara's similarity to herself (before she interacted with LO87) is also calculated. As she learns, she will become less similar to the her past assessments (and these past assessments will be used to help determine if she should experiment a learning object a second time). Barbara is calculated to be highly similar to her past assessment (0.97) since the learning object randomly assigned to her did not benefit her very much. This makes her highly similar to her previous assessment, which means the experience she had with the learning object will be given a higher weight. Since that experience was neutral, the consequence will be to push her towards other learning objects where less similar peers received higher benefit. Barbara is assessed again (line 11) and using this assessment in the CLA the highest predicted benefit is for LO42 (which Adam had a beneficial interaction with). The CLA recommends both LO42 and LO101 equivalently, since both have the exact same interaction history. One of the two is randomly assigned (LO101 in this case) and Barbara begins her interaction (line 16). After Barbara completes her interaction, the two learning objects will no longer receive equivalent predicted benefits (because LO101 will have Barbara's interaction as well as Adam's). Annotations which Adam left are determined to be shown to Barbara (line 15) since they are the only options for annotations currently attached to this Learning Object. These annotations are the ones which were attached to the sections that Adam chose to keep when he divided the learning object. Algorithm 25 loops while Barbara is interacting with the learning object, and Barbara rates many of the annotations Adam left (line 17), liking some and disliking others. She also may create new annotations (line 18) which would only be attached to LO101, not to LO42.

6.5 Pushing Examples Through Driver

We now focus on the mathematical formulas behind the algorithms previously presented and push a number of students and learning objects through multiple interactions. We present the student evaluations as a single-dimension of knowledge for ease of reading.

Students S1, S2, S3 have used Learning Objects L1, L2, L3 with Annotations A1-A6 attached to them. The students have the results of their last post-test assessment indicated in the square brackets next to their identifier. L3 has been divided (by S2 after a previous interaction) and the new objects created L3-a during this division has lost A3, which was attached to a part that was cut out. Student S4 is newly introduced to the system. All annotations were left by previous students, with the default reputation (0.5). Reputations are calculated using the Trust-Based approach (described in Algorithm 6, using an N_{min} of 10)

Students	Lessons	Annotations
S1[A-]	L1[S1,B+,A-][S2, C-, C-]	A2[S1↑,S2↑], A4[S2↓]
S2[C-]	L2[S1,B+,B+][S3, B+,B]	A1[S1↓,S3↑]
S3[B]	L3[S2,C,C-]	A3, A5[S2↑], A6
S4[C+]	L3-a[S2,C,C-]	A5-a[S2↑], A6-a

↑ is voted up, ↓ is voted down

We compute similarities between students, of use in determining the next learning object to show. This is done by computing the difference between the current assessments of each student, for example S1 and S2 have a difference of 6 because A- is 6 letter grades above C-. We also initialized the reputation of each student to be 0.5 and calculate the reputation of each annotation.

The two weights (for example, 0.2 and 0.8 in the reputation of A2 below), sum to unity and represent what importance to place on the community reputation (the first number, 1 below) and what importance to place of the annotator's reputation (the second 0.5 below). These weights are determined by the number of votes that have been left on an annotation, for example 0.1 and 0.9 after 1 vote, 0.2 and 0.8 after 2 votes, 0.3 and 0.7 after 3 votes, etc. The first 1 comes from the fact that all voters have liked the annotation (i.e. ($votes_{for}$ - $votes_{against}$) / $total_{votes}$ = (2 + 0) / 2 = 1). The 0.5 is the reputation of the annotator (some student outside of this example) who has the default (starting) reputation.

similarity(S1,S2) = 1 / (1 + 6) = 0.143	reputation(A1) = 0.2(0.5) + 0.8(0.5) = 0.5
similarity(S1,S3) = 1 / (1 + 2) = 0.333	reputation(A2) = 0.2(1) + 0.8(0.5) = 0.6
similarity(S1,S4) = 1 / (1 + 4) = 0.2	reputation(A3) = 0(0.5) + 1(0.5) = 0.5
similarity(S2,S3) = 1 / (1 + 4) = 0.2	reputation(A4) = 0.1(0) + 0.9(0.5) = 0.45
similarity(S2,S4) = 1 / (1 + 2) = 0.333	reputation(A5) = 0.1(1) + 0.9(0.5) = 0.55
similarity(S3,S4) = 1 / (1 + 2) = 0.333	reputation(A6) = 0(0.5) + 1.0(0.5) = 0.5
	reputation(A5-a) = 0.1(1) + 0.9(0.5) = 0.55
	reputation(A6-a) = 0(0.5) + 1.0(0.5) = 0.5
	reputation(S1) = 0.5 // default
	reputation(S2) = 0.5 // default
	reputation(S3) = 0.5 // default
	reputation(S4) = 0.5 // default

The CLA calculations are as below. The kappa (κ) is the normalizing factor, as discussed for Equation 3.1. The purpose of this value is to normalize all experiences attached to a learning object. In the first calculation below, predBenefit(L1, S1), there are 2 previous experiences attached; therefore, the κ has a value of $\frac{1}{2}$. In the parentheses, there are two terms. Each term is made up of a similarity multiplied by a benefit. The similarity is calculated as explained above. The benefit is the absolute difference between the student's pre-test assessment and post-test assessment (as explained in Chapter 3).

predBenefit(L1, S1) = $\frac{1}{2}$* (0.5 * 1/18 + 0.143*0) = 0.0278
predBenefit(L2, S1) = $\frac{1}{2}$* (0.5 * 0 + 0.5 * -1/18) = -0.0139
predBenefit(L3, S1) = 0.2 * -1/18 = -0.0111
predBenefit(L3-a, S1) = 0.2 * -1/18 = -0.0111

predBenefit(L1, S2) = $\frac{1}{2}$* (0.2 * 1/18 + 1.0*0) = 0.00556
predBenefit(L2, S2) = $\frac{1}{2}$* (0.2 * 0 + 0.2 * -1/18) = -0.00556
predBenefit(L3, S2) = 0.5 * -1/18 = -0.0278
predBenefit(L3-a, S2) = 0.5 * -1/18 = -0.0278

predBenefit(L1, S3) = $\frac{1}{2}$* (0.5 * 1/18 + 0.2*0) = 0.0139
predBenefit(L2, S3) = $\frac{1}{2}$* (0.5 * 0 + 0.5 * -1/18) = -0.0139
predBenefit(L3, S3) = 0.25 * -1/18 = -0.0139
predBenefit(L3-a, S3) = 0.25 * -1/18 = -0.0139

predBenefit(L1, S4) = $\frac{1}{2}$* (0.25 * 1/18 + 0.333 * 0) = 0.00694
predBenefit(L2, S4) = $\frac{1}{2}$* (0.25 * 0 + 0.25 * -1/18) = -0.0069
predBenefit(L3, S4) = 0.5 * -1/18 = -0.0278
predBenefit(L3-a, S4) = 0.5 * -1/18 = -0.0278

Each student would be assigned the most beneficial learning object for the next step of their learning. → S1 is assigned to L1 again (and goes from an A- to an A assessment) and we assume that he leaves annotation A7, S2 also gets assigned to L1 (and goes from a C- to a D+), rates A7 (thumbs up) and leaves annotation A8. S3 gets assigned to L1 (and stays a B), rates annotations A2, A7 and A8. A4 isn't shown due to its low reputation according to the personalized calculation shown below (using a value of 3 for X, the number of annotations to be shown). S3 leaves annotation A9. Even though the predicted benefit for S4 is highest for L1, since this student is newly introduced to the system, he gets randomly assigned for his first experience and is randomly assigned to L2 (and goes from a C+ to a B+) rates A1 and leaves annotation A10. S2 suggests a subdivision of L1, which includes the portions covered by Annotations A4 and A8 (which are copied over).

We need to determine which annotations to show to a student; these calculations are shown below. We are deciding which 3 of A2, A4, A7 and A8 to show to student S3. We need to first examine the annotation rating similarity between student S3 and the other students who have rated these annotations, students S1 and S2. We look at annotations that the students have mutually rated (for example, S3 and S1 have both rated A1 and because their votes are different their similarity is -1). This similarity is used to modify a vote. Because S1 and S3 have a similarity of -1 then effectively S1's vote on annotation A2 is removed from the total. The total community vote is then based exclusively on the vote left by S2 who is neither similar nor dissimilar to S3.

showAnnotation(S3,L1)

similarityAnn(S3,S1) = -1
similarityAnn(S3,S2) = 0

calculate personalized reputations
rep(A2) = 0.2 (1) + 0.8 (0.5) = 0.6 // 1 vote for from S2, 0 votes for from S1
rep(A4) = 0.1 (0) + 0.9 (0.5) = 0.45 // 1 votes against from S2
rep(A7) = 0.1(1) + 0.9(0.5) = 0.55 // 1 for for from S2
rep(A8) = 0(0.5) + 1(0.5) = 0.5 // no votes recorded

Therefore, the top 3 annotations are shown: A2, A7 & A8

We now have the following picture for our students, learning objects and annotations. Since students S1-S4 have now left annotations which have been rated by other students, we can now calculate a reputability for each student as an annotator, using the average reputation of all annotations they have left.

Students	Lessons	Annotations
S1[A]	**L1**[S1,B+,A-][S2, C-, C-][S1,A-,A]	A2[S1↑,S2↑,S3↓], A4[S2↓],
S2[D+]	[S2,C-,D+][S3,B,B]	A7[S2↑,S3↑], A8[S3↑], A9
S3[B]	**L2**[S1,B+,B+][S3, B+,B][S4,C+,B+]	A1[S1↓,S3↑,S4↓],A10
S4[B+]	**L3**[S2,C,C-]	A3, A5[S2↑], A6
	L3-a[S2,C,C-]	A5-a[S2↑], A6-a
	L1-a[S1,B+,A-][S2, C-, C-][S1,A-,A]	A4-a[S2↓], A8-a[S3↑]
	[S2,C-,D+][S3,B,B]	

↑ is voted up, ↓ is voted down

similarity(S1,S2) = 1 / (1 + 8) = 0.111	reputation(A1) = 0.3(0.333) + 0.7(0.5) = 0.450
similarity(S1,S3) = 1 / (1 + 3) = 0.25	reputation(A2) = 0.3(0.667) + 0.7(0.5) = 0.550
similarity(S1,S4) = 1 / (1 + 2) = 0.333	reputation(A3) = 0(0) + 1(0.5) = 0.5
similarity(S2,S3) = 1 / (1 + 5) = 0.167	reputation(A4) = 0.1(0) + 0.9(0.5) = 0.45
similarity(S2,S4) = 1 / (1 + 6) = 0.143	reputation(A5) = 0.1(1) + 0.9(0.5) = 0.55
similarity(S3,S4) = 1 / (1 + 1) = 0.5	reputation(A6) = 0(0) + 1.0(0.5) = 0.5
	reputation(A5-a) = 0.1(1) + 0.9(0.5) = 0.55
	reputation(A6-a) = 0(0.5) + 1.0(0.5) = 0.5
	reputation(A7) = 0.2(1) + 0.8(0.5) = 0.6
	reputation(A8) = 0.1(1) + 0.9(0.5) = 0.55
	reputation(A9) = 0(0.5) + 1.0(0.5) = 0.5
	reputation(A10) = 0(0.5) + 1.0(0.5) = 0.5
	reputation(A4-a) = 0.1(0) + 0.9(0.5) = 0.45
	reputation(A8-a) = 0.1(1) + 0.9(0.5) = 0.55
	rep(S1) = average(rep(A7)) = 0.6
	rep(S2) = average(rep(A8),rep(A8-a)) = 0.55
	rep(S3) = average(rep(A9)) = 0.5
	rep(S4) = average(rep(A10)) = 0.5

predBenefit(L1, S1) = 1/5 * (0.333*2/18 +0.125*0+0.5*1/18+0.125*-1/18+0.25*0) = 0.0116
predBenefit(L2, S1) = 1/3 * (0.333 * 0 + 0.333 * -1/18 + 0.167*3/18) = 0.00311

predBenefit(L3, S1) = 0.143*-1/18 = -0.00794
predBenefit(L3-a, S1) = 0.143*-1/18 = -0.00794
predBenefit(L1-a, S1) = 0.0116 // same history as L1

predBenefit(L1, S2) = 1/5 * (0.143*2/18 +0.5*0+0.125*1/18+0.5*-1/18+0.167*0) = -0.000989
predBenefit(L2, S2) = 1/3 * (0.143 * 0 + 0.143 * -1/18 + 0.25*3/18) = 0.0112
predBenefit(L3, S2) = 0.333*-1/18 = -0.0185
predBenefit(L3-a, S2) = 0.333*-1/18 = -0.0185
predBenefit(L1-a, S2) = -0.000989 // same history as L1

predBenefit(L1, S3) = 1/5 * (0.5*2/18 +0.2*0+0.333*1/18+0.2*-1/18+1*0) =0.0126
predBenefit(L2, S3) = 1/3 * (0.5 * 0 + 0.5 * -1/18 + 0.333*3/18) = 0.00924
predBenefit(L3, S3) = 0.333*-1/18 = -0.0185
predBenefit(L3-a, S3) = 0.333*-1/18 = -0.0185
predBenefit(L1-a, S3) = 0.0126// same history as L1

predBenefit(L1, S4) = 1/5 * (1*2/18 +0.167*0+0.5*1/18+0.167*-1/18+0.5*0) =0.0259
predBenefit(L2, S4) = 1/3 * (1 * 0 + 1 * -1/18 + 0.25*3/18) = -0.00463
predBenefit(L3, S4) = 0.2*-1/18 = -0.0111
predBenefit(L3-a, S4) = 0.2*-1/18 = -0.0111
predBenefit(L1-a, S4) = 0.0259 // same history as L1

→ S1 has the highest predicted benefit from L1 and L1-a and gets randomly assigned to L1 again (and goes from an A to an A-), S2 gets assigned to L2 (and goes from a D+ to a C+) and leaves annotation A11, S3 has the highest predicted benefit from L1 and L1-a and gets randomly assigned to L1-a (and goes from a B to a B-) and S4 has the highest predicted benefit from L1 and L1-a and gets randomly assigned to L1-a (and goes from a B+ to an A-) and leaves annotation A12. Various annotations are rated. Annotation 11 is left by S2 on L2 and annotation 12 is left by S1 on L1.

We now have to determine which annotations from A2, A4, A7, A8, A9 are to be shown to S1. The calculations are below.

showAnnotation(S1,L1)

similarityAnn(S1,S1) = 1 // each student has perfect similarity with themselves.
similarityAnn(S1,S2) = 1 (agreed once, never disagreed)
similarityAnn(S1,S3) = -1 (never agreed, disagreed twice)
similarityAnn(S1,S4) = 1 (agreed once, never disagreed)

When two students are similar, the contribution from the voting student is emphasized, as detailed in lines 8 and 10 in Algorithm 20, and its contribution to the community reputation is enhanced.

For S1:
calculate personalized reputations:

rep(A2) = 0.3 (1) + 0.7 (0.5) = 0.65 // 2 votes for from self, 2 votes for from S2, 0 votes from S3

rep(A4) = 0.1 (0) + 0.9 (0.5) = 0.45 // 2 votes against from S2

rep(A7) = 0.2 (1) + 0.8(0.5) = 0.6 // 2 votes for from S2, 0 votes against from S3

rep(A8) = 0.1(0.5) + 0.9(0.5) = 0.5 // 0 votes for from S3

rep(A9) = 0(0.5) + 1(0.5) = 0.5 // not votes

Therefore, the top 2 annotations are shown: A2 & A7. Since A8 & A9 tied for third ranking, one of them is randomly shown - A9 in this case.

Students	Lessons	Annotations
S1[A-]	**L1**[S1,B+,A-][S2,C-,C-][S1,A-,A]	A2[S1↑,S2↑,S3↓], A4[S2↓],
S2[C+]	[S2,C-,D+][S3,B,B][S1,A,A-]	A7[S1↑,S2↑,S3↑], A8[S3↑],
S3[B-]	**L2**[S1,B+,B+][S3,B+,B][S4,C+,B+]	A9[S1↑],A12
S4[A-]	[S2,D+,C+]	A1[S1↓,S2↑,S3↑,S4↓],
	L3[S2,C,C-]	A10[S2↓],A11
	L3-a[S2,C,C-]	A3, A5[S2↑], A6
	L1-a[S1,B+,A-][S2,C-,C-][S1,A-,A]	A5-a[S2↑], A6-a
	[S2,C-,D+][S3,B,B][S3,B,B-][S4,B+,A-]	A4-a[S2↓,S3↑,S4↑],A8-a[S3↑,S4↑]

↑ is voted up, ↓ is voted down

similarity(S1,S2)	1 / (1 + 7) = 0.125
similarity(S1,S3)	1 / (1 + 3) = 0.25
similarity(S1,S4)	1 / (1 + 2) = 0.333
similarity(S2,S3)	1 / (1 + 5) = 0.167
similarity(S2,S4)	1 / (1 + 6) = 0.143
similarity(S3,S4)	1 / (1 + 1) = 0.5

reputation(A1) = 0.4(0.5) + 0.6(0.5) = 0.5
reputation(A2) = 0.3(0.667) + 0.7(0.5) = 0.550
reputation(A3) = 0(0.5) + 1(0.5) = 0.5
reputation(A4) = 0.1(0) + 0.9(0.5) = 0.45
reputation(A5) = 0.1(1) + 0.9(0.5) = 0.55
reputation(A6) = 0(0.5) + 1.0(0.5) = 0.5
reputation(A5-a) = 0.1(1) + 0.9(0.5) = 0.55
reputation(A6-a) = 0(0.5) + 1.0(0.5) = 0.5
reputation(A7) = 0.3(1) + 0.7(0.5) = 0.65
reputation(A8) = 0.1(1) + 0.9(0.5) = 0.55
reputation(A9) = 0.1(1) + 0.9(0.5) = 0.55
reputation(A10) = 0.1(0) + 0.9(0.5) = 0.45
reputation(A4-a) = 0.3(0.667) + 0.7(0.5) = 0.550
reputation(A8-a) = 0.2(1) + 0.8(0.5) = 0.6
reputation(A11) = 0(0.5) + 1.0(0.55) =0.55
reputation(A12) = 0(0.5) + 1.0(0.6) = 0.6

rep(S1) = avg(rep(A7),rep(A12)) = 0.625
rep(S2) = avg(rep(A8),rep(A8-a),rep(A11)) = 0.567
rep(S3) = avg(rep(A9)) = 0.55
rep(S4) = avg(rep(A10)) = 0.45

Chapter 7

Additional Validation: User Study

In Chapters 3-6 we validated our models using simulations. To learn more about the effectiveness of our approach with real users we also conducted a preliminary evaluation with participants at the University of Waterloo. We chose as an application domain enabling users to learn about how to care for a child with autism (which may arise as a home healthcare scenario, of interest to projects such as hSITE [63], with which we are involved). This study was focused on providing additional validation for our curriculum sequencing algorithm. It also offered some insights into the value of our annotations approach and our proposal for corpus division.

7.1 Overview

Our first step was to assemble a repository of learning objects: the material that our users would learn from. In collaboration with a clinical psychologist specializing in children and autism, we created 20 learning objects (16 text articles and 4 videos) that each took about 5 minutes to experience. Also in collaboration with the psychologist we created a 10-question multiple choice assessment, covering material from the learning objects. This was used to carry out the pre- and post-test assessments which serve to model learning gains in students (and form a component of our algorithm for determining which objects to present to each student).

We hypothesized that a group of students using our peer-based technique for selecting learning objects would show greater learning gains than a control group that had learning objects randomly assigned to them. The aim of our study, therefore, was to validate our proposed Collaborative Learning Algorithm for curriculum sequencing (Algorithm 1) – the centerpiece of our overall peer-based learning framework.

A qualitative component of this experiment evaluated participants' feelings towards the concept of annotations by showing them example learning objects with annotations attached to them. This survey is shown in Appendix C[1].

In order to obtain feedback about corpus division, we also explained our corpus approach to participants and then offered them the opportunity to subdivide each learning object that they were shown, as the learning proceeded.[2] This survey was shown to the participants after each learning object and is shown in Appendix B. We finally obtained more information during an exit survey (Appendix C) where participants responded to questions asking them how they felt about this option of streamlining learning objects. The entire session lasted approximately 1 hour. 23 participants were involved in our experiment, including graduate students, undergraduate students and staff members at the university. All were at least 18 years old, fluent in English and not an expert in autism spectrum disorders.

7.2 Curriculum Sequencing

The focus of our study was validating our proposed curriculum sequencing algorithm.

7.2.1 Procedure

To assemble our repository of learning objects we worked in conjunction with a clinical psychologist. Initially she provided extensive, psychologically sound articles and videos about the care of children with autism spectrum disorder. From this, we distilled information to what we felt would be 5 minute lessons[3] that were either taken from text or from much longer lecture videos. These 5 minute streamlined versions were shown again to the psychologist to ensure that they still presented sound information after being taken out of

[1] We also asked in this exit survey for students to indicate which learning objects they would have liked to be assigned.

[2] These subdivisions were not used by other participants. Getting enough data, with the limited number of participants, to differentiate between original learning objects and streamlined versions would have been challenging.

[3] After creating a rough cut of the lesson, I timed myself reading through them and looked at the timestamps on the video to verify that they took approximately 5 minutes each.

their larger context. Eventually the psychologist also ensured that sufficient information was present in these lessons to answer the assessment questions developed.

In recruiting participants for this study we first intended to bring on board participants connected to the hSITE project whom we knew as parents of children with autism. As this path was not entirely successful, we then focused instead on obtaining participants who were students or staff at the University of Waterloo. The recruitment letter requesting participation is presented in Appendix D; it clearly indicates that we were excluding anyone with significant expertise in autism spectrum disorders. In fact, we eliminated someone who volunteered, upon learning that they had received training to work with children with autism.

We decided to begin with just two participants in order to troubleshoot our experimental approach, proposed set of learning objects and assessment questions. We had in mind that these students would receive a random set of learning objects with each subsequent learning object not being influenced by their experience with the previous object. We had at that time a proposed assessment quiz that was administered before and after each learning object (the same assessment quiz each time). This quiz was a set of very general questions about autism that was not tied to the learning objects specifically. We observed that the two participants improved on their assessments, even though subsequent learning objects did not offer information that served to answer the questions on which they had improved. We learned that the participants were in fact improving because of the assessment quiz itself (using information from one question to help answer another question). From here we interacted with two psychologists experienced in research methods and concluded that it was important for us to ensure that our assessment quiz was much more directly tied to the learning objects that would be experienced. We then revised our assessment quiz and assembled a group of participants for our study, using this revised quiz for assessing these participants.

Each participant experienced 5 learning objects and was assessed before and after each for a total of 6 assessments. The assessments were the same 10 multiple choice questions each time[4]. This was done in part to ensure that we were modeling comparable learning experiences from the participants. Note that we considered, and rejected, the idea of using different assessments, counter-balanced using a Latin square, for two reasons. First, our primary evaluation of each participants session would be their learning gain, namely final assessment (post-test) minus initial assessment (pre-test). If our experiment used a variety of assessment for each learning object we would be "comparing apples to oranges" if we

[4] The first assessment, before the participants have experienced any learning objects, measures their initial knowledge about the subject at the beginning of the session.

used different versions of the assessments for pre- and post- tests. Having a separate pre- and post- test would be one possibility, but there was a concern that having too many assessments might overwhelm participants. Secondly, our technique benefits greatly from data on each learning object. If we used different assessments this would result in having a fraction of the data points on each learning object, leading to a sparsity issue.

In the end, our quiz was designed so that each question was covered well by different learning objects in the repository (and more than one learning object served to help a student to respond to that question). See Appendix C for a description of the repository of learning objects and Appendix A for the assessment quiz that was used.

After experiencing each learning object, each participant did the assessment quiz[5] and also answered a separate questionnaire allowing the student to propose a streamlining (division) of that learning object. At the end of the experiment each participant was given an exit survey asking them their overall feelings about streamlining and soliciting general feedback.

We decided at the outset that we would use a control group, who experienced learning objects not using our techniques, and a treatment group, whose learning objects were derived by our techniques, in order to contrast the two groups. We hypothesized that the curriculum sequencing approach would lead to greater learning gains in the treatment group. The first 12 participants were randomly assigned learning objects each time during their session. They were used both as a control group and to provide training data for our technique. The next 11 participants experienced a curriculum sequence provided by our approach.

Participants read hardcopy articles or watched videos on a provided netbook as a "Wizard of Oz" style study was performed (see Section 2.6.5). For our technique, a program was written using the CLA (Algorithm 1) and the answers provided by participants in their pre-test assessments served as the current student assessment; a new recommendation for a learning object was then determined. This sequence continued until the student had experienced five different learning objects. In essence, the first 12 participants served to prime the system for the remaining participants. After this phase, each learning object in the repository had 3 experiences recorded: while the initial control group of students were shown a random set of objects, which objects would be presented to each was determined off-line in a way that ensured that each object would be shown to 3 different participants[6].

[5] Note that we required a very strong impact from the learning object on the assessment in this study, since each learning object was only 5 minutes long and the total instruction during the experiment was 25 minutes; our techniques in general can function with less specific assessments, given a far greater amount of student data - i.e. longer instruction time and more students.

[6] For the 12 participants, three random lists of all integers between 1 and 20 were created. The first

The net-benefit obtained by each subject in the control group (number of questions correct between pre and post-test) became part of that object's interaction history. As a result, each learning object in the repository acquired an interaction history with exactly 3 entries. This then ensured that the students in the treatment group were experiencing a sufficiently rich collaborative learning algorithm. For the participants in our experimental group, determining the similarity between the current student and previous peers was measured by comparing the number of questions on the assessment that were answered identically. Only the data collected from the training group was used to make recommendations to the experimental group.[7] No learning objects were shown twice to the same participant.

My role as "wizard" was to first of all take the very first pre-assessment quiz completed by the subject and enter their answers to each question into a program on the netbook. This produced a list of 5 learning objects with the highest predicted benefit (according to the CLA) in order. The most preferred learning object that the student had not previous seen was either handed to the student as a print out, or the video was loaded on the netbook and shown to the participant. Note that we made an executive decision not to show the same learning object twice; this was reasonable, given the very brief length of the learning objects and of the total instructional experience. I repeated this same procedure after each student had completed an assessment.

More details on how the calculations were done when determining which learning objects to show each student are presented in Appendix E. This, in fact, involved a multi-dimensional model of knowledge and students were similar if they did well on the same questions of the quiz.

7.2.2 Results

We first compared the learning gains of our 11 experimental group participants, namely the post-test (their final assessment) minus the pre-test (their first assessment).

These results can be interpreted that, on average, participants in the control group got 1.83 more questions correct (out of 10) after completing the 5 learning objects and

participant saw the first 5 entries on the first list, the second participants saw entries 6-10 on the first, etc. with the 12th participant seeing the last 5 entries on the third list. This ensured that 1) every student was assigned 5 distinct random learning objects and 2) that each learning object had data from 3 separate students.

[7] Had we followed our proposed approach and continually added data for the program to make recommendations, the final participants would have been given learning objects based on a richer repository of data and the experimental group would not have been provided with a consistent treatment.

	Mean	s.d.	Mean (without P20)	s.d. (without P20)
Control	1.83	1.27		
Experimental	3.09	2.21	3.4	2.07

Table 7.1: Comparison of overall learning gains of users in control and experimental groups

participants in the experimental group got an average of 3.09 more questions correct (see Table 7.1).

P20 was a participant who did not seem to be taking the experiment seriously, did not read learning objects fully and rushed through the experiment (finishing in about 40 minutes when most participants took about 1 hour). The data was analyzed with and without this participant's data included.

The results were statistically reliable at p=0.059 (one-sided, two samples, unequal variance t-test) which was not statistically significant. With participant 20 removed, the results were statistically reliable at p=0.027 (one-sided, two samples, unequal variance t-test) which was statistically significant.

Next, we compared the proportional learning gains of participants (see Table 7.2). This was to take into consideration the suggestion of Jackson and Graesser [37] that simple learning gains are "biased towards students with low pretest scores because they have more room for improvement". This is measured using $\frac{posttest-pretest}{10-pretest}$[8].

	Mean	s.d.	Mean (without P20)	s.d. (without P20)
Control	0.530	0.452		
Experimental	0.979	1.07	1.08	1.02

Table 7.2: Comparison of proportional overall learning gains of users in control and experimental groups

The results were statistically reliable at p=0.10 (one-sided, two samples, unequal variance t-test) which was not statistically significant. With participant 20 removed, the results were statistically reliable at p=0.071 (one-sided, two samples, unequal variance t-test) which also was not statistically significant.

Next, we considered the per-LO learning gains of each student (see Table 7.3). Here, the change in assessment after assignment of a single learning object, were measured for each

[8] 10 is the maximum possible score on a 10 question multiple-choice quiz.

learning object experienced and the average computed. This average was then compared for the control and experimental groups.

	Mean	s.d.	Mean (without P20)	s.d. (without P20)
Control	0.367	0.253		
Experimental	0.618	0.442	0.68	0.413

Table 7.3: Comparison of average learning gains of users in control and experimental groups

The results were statistically reliable at p=0.059 (one-sided, two samples, unequal variance t-test) which was not statistically significant. With participant 20 removed, the results were statistically reliable at p=0.027 (one-sided, two samples, unequal variance t-test) which was statistically significant.

Taken together, our results indicate that students presented with learning objects determined by our algorithm achieved greater learning gains than those who were randomly assigned objects.

7.2.3 Annotations: Qualitative Results

Since our overall motivation was to eventually offer our system to real users, in order to learn more about its usability and user satisfaction with the results that it delivers, we explored out participants' reaction to our other central techniques - annotations and corpus division. We have already presented results that validate our particular intelligent tutoring algorithm that drives the selection of the learning objects (against which annotations are applied) but once the learning session with each student was complete, we also showed our participants a mock-up of annotations attached to learning objects and asked our participants 4 questions about the annotations as part of an exit survey:

1. Do you find any value in using learning objects with annotations?
2. How likely would you be to contribute an annotation to a learning object if using a system that supported this?
3. How often might you leave annotations?
4. How satisfied would you be reading annotations left by previous students?

Participants were given a 11 point scale, ranging from -5 to 5 for Q1, Q2 and Q4 with the labels “less value”, “unlikely” and “unsatisfied” (respectively) at -5, “neutral” at 0 and “more value”, “likely” and “satisfied” (respectively) at 5. For Q3 participants were given an 11 points scale ranging from 0 to 10 with the labels “never” at 0 and “always” at 10. For the 23 participants the feedback was (question, mean, standard deviation):

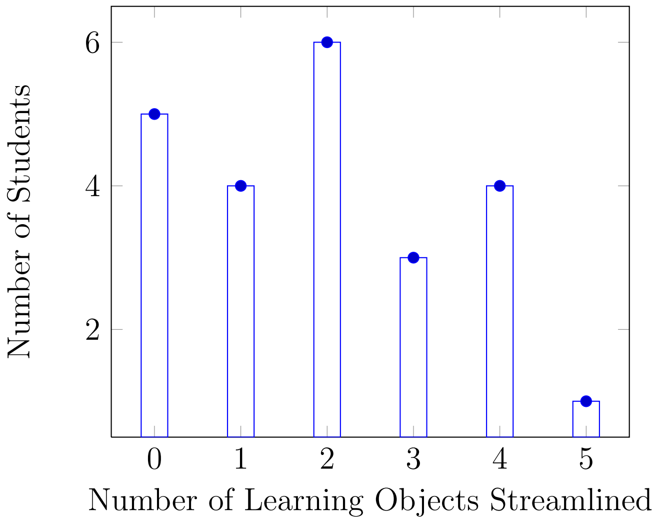
Question	Mean	s.d.
Q1	2.5	2.79
Q2	1.63	3.21
Q3	4.64	2.59
Q4	2.32	2.98

Table 7.4: Mean answer values to annotation survey questions

Although participants were mostly neutral with respect to creating new annotations (Q2,Q3), they were clearly positive about using a system where other students have left annotations on learning objects for them.

7.2.4 Corpus Approach: Qualitative Results

While the students were progressing with their learning, after each participant had experienced a learning object and completed the post-test assessment and questionnaire about the learning object, they were then invited to optionally suggest a streamlined version of the learning object. The first time they were asked to do this, our approach to corpus-division was explained to them. This part of our human study was to determine possible interest in corpus division from users.



In spite of being told that it was up to them whether or not to streamline learning objects, only 5 out of 23 participants declined to streamline any objects. On average, participants suggested streamlined versions for 2 of the 5 learning objects they saw.[9]

Each participant was asked 3 questions about the corpus approach during their exit survey:

1. How would you rate the difficulty of creating a new streamlined learning object?
2. How would you rate the difficulty of deciding what content to include in a streamlined version?
3. How would you rate the usefulness of a system offering a user the full version or streamlined version of content like you've seen?

Participants were given a 11 point scale, ranging from -5 to 5 with the labels "difficult" at -5, "neutral" at 0, and "easy" at 5 for Q1 and Q2 and "useless" at -5, "neutral" at 0 and "useful" at 5 for Q3.

For the 23 participants the feedback is provided in Table 4.

Although participants were mostly neutral with respect to creating streamlined versions of learning objects (Q1 and Q2), they were clearly positive about using a system where other students create streamlined learning objects for them. This conforms to research

[9] In practice, participation may be lower if there isn't a researcher sitting across the table when students are deciding whether or not to streamline; however there was clearly a willingness to engage in this activity.

Question	Mean	s.d.
Q1	0.227	3.35
Q2	0.864	2.949
Q3	3.773	1.232

Table 7.5: Mean answer value to exit survey questions

on participatory culture [4] which has shown that consumers usually greatly outnumber contributors. It has been shown to be possible [18] to use incentives to encourage greater participation.

Chapter 8

Discussion

In this chapter we reflect on some of the derived benefits of our research determined by reflecting generally on the model as presented in Chapters 3-6. We return to chronicle the central contributions and to highlight key design decisions in Chapter 9.

8.1 Contributions to ITS Construction

Intelligent tutoring systems have long been shown to have large costs of development. The consequence of this is that approaches which have been demonstrably effective in the laboratory do not make it to the classroom. This work provides a number of techniques to decrease the cost of constructing an intelligent tutoring system. While there is no "silver bullet" to make construction of an ITS cheap, fast and easy it is hoped that work such as this will remove some of the barriers to deploying ITS.

The curriculum sequencing approach provides an alternative to manually setting curricula for specific students or groups of students. As well as being adaptive and personalized, it automatically adjusts itself to new information and possible curriculum choices that even the most attentive classroom teacher would be unable to provide. In contrast to ITS techniques of using ontologies (also known as taxonomies or metadata) to describe learning objects, perform student modeling on all users and constructing elaborate reasoning about constraints to match learning objects to students our systems automatically interprets existing data of interactions to make recommendations. Ontologies in particular have repeatedly been shown to be problematic in practice, as it is time consuming to provide full specifications for all objects in a repository, they quickly get out of date when objects change or new objects are introduced, and consumers of objects develop new ontology categories they want objects marked up with.

For a content developer with limited time to devote to adjusting the learning material, our annotation and corpus approaches allow motivated students to have the experience of adjusting the learning material to better suit the student population they are a part of. While studies have shown [4] that participatory culture like Wikipedia and blogs have a low participation rate, it is also important to note that those who do participate are voluntarily offering to make adjustments to the material they are consuming. This is a powerful source of cheap labour that can be used to enhance ITS and the learning outcomes of students who use them.

8.2 Approaches to the Cold Start Problem

The cold start problem is a broad issue that occurs whenever a technique requires information in order make decisions but doesn't have the required information. This is a particular vexing problem in the context of recommender systems, where new users reasonably want a recommendation but the system has no information about them to base a recommendation on.

Various approaches have been offered to deal with this problem, and in this work we provide 3 techniques in the intelligent tutoring systems curriculum sequencing domain and 3 techniques in the trust modeling domain. Each of these approaches can be extrapolated to be used for other problems.

In curriculum sequencing we first offer the raw ecological approach which deals with the cold start problem by making 3 random recommendations to each new user, then making deliberate recommendations after this. The number 3 is a flexible parameter that can be adjusted. We saw useful results from this approach which is appealing in its simplicity to implement and faithfulness to the recommendation approach (after the 3 random "seeds" it follows the algorithm precisely). The ecological with pilot uses a portion of the user base to deliberately explore rather than trying to make worthwhile recommendations. The data that is obtained from this is very useful when the remainder of the students participate in the system. Real world scenarios where a portion of the user base can be sacrificed in this manner may be rare. Examples could include a commercial system that would allow free access to users who would be given an exploratory role in the system (and be given basically random recommendations) and paid users who would benefit from the data provided by the free users. Finally, the simulated annealing approach has a decreasing amount of randomness that persists for the first half of the course of study then becomes wholly deterministic. The pilot and simulated annealing approach both proved to perform better in an environment with more choices, which makes sense since they have a greater focus on exploration. The price of this randomness on real human students has not been examined, and it can be imagined that in certain scenarios having a random assignment of recommendations through the first half of a course of study may be a undesirable due to the frustration that students may experience.

For our trust modeling approach, we allowed students to rate annotations they found useful, then showed annotations to students based on what other, similar students had found useful in the past. The first approach to this, the Tally approach, showed the annotations which had the highest number of positive ratings compared to total ratings. The second approach used a Cauchy CDF to incorporate both the Tally and the overall reputation of the student who left the annotation. The third approach, Trust-Based, used a weighting based on the number of ratings an annotation had to gradually shift the recommendations from being based on the reputation of the student who left the annotation to being based on the ratings left on that annotation. Each of these approaches showed promising results in recommendation of annotations. The Cauchy and Trust-Based used a novel approach of incorporating the inherent reputation of the person creating the annotation as a way of dealing with the cold start problem: for new annotations, rather than reasoning about the annotation itself, we reason about the student who left it.

8.3 The Role of Assessment

Consider a system that endeavours to convey scientific skepticism to students. As part of this process, students are assigned an article written in favour of therapeutic touch (TT)[1]. Although the article's content is actually harmful, presenting misinformation, it can have a positive educational impact on students by providing an example of something to be skeptical about. This impact could be particularly pronounced if this article was paired with a detailed analysis of the claims in the original article, or general information about evaluating health information.

Our system, which assigns learning objects and annotations based on assessments, doesn't give recommendations based on the validity of educational content of learning objects and, in fact, does not perform any reasoning about this. Instead, recommendations are pragmatically made based on what has helped a student learn. Whatever pedagogical

[1] A therapy that involves healing by laying on hands which was debunked by 9 year old Emily Rosa [69].

approach or specific learning objects facilitate this for a student will be more likely to be shown to similar students in the future. Therefore, this deflects the need to explicitly screen every learning object in the repository in order to remove the ones with weaker educational value. The system itself will identify the stronger and weaker learning objects.

We note that although we focus on assessment and performance, this is not to the detriment of overall student learning. We are focused at a broader level at determining what are good and bad learning objects in order to show those objects with the best predicted benefit. In the context of our work, "good" simply refers to learning objects that have improved the assessments of students, while "bad" refers to learning objects that have not (or which have harmed the assessment of students). While this, by necessity, places a burden on correctly assessing students, when validating our approach we allowed for some errors in assessment by adding a noise factor (Section 3.6) and found that with a sufficiently large amount of student data on which to base recommendations, our models were highly robust to problems with the assessment.

8.4 Simulated Students

8.4.1 Use of Simulation For Validation

In addition to being expensive to develop, ITS are expensive to evaluate [80]. The simulation portion of our work should be viewed as an overview for an approach to perform an inexpensive first evaluation of an ITS using a simulation. As such, our work serves to demonstrate the feasibility and the value of using simulated students in the design of an ITS.

There are a number of benefits to using simulations to validate ITS. Studies of human students are difficult and time consuming to conduct. A simulation of a group of students that was capable of being run on a typical desktop computer in 30 minutes might take months or years to conduct with human participants. Human ethics approval and participant recruitment increases the cost of conducting such experiments further. Research focusing on the education of children has stricter ethical requirements than with adults. It is far more difficult to control the environment that human learners are educated within.

Some might question how we could convince a group of real students to suffer through matches with inappropriate learning objects in order to get to the point where better matches can be made. In part, this is one of the benefits offered by our simulated environment: that the students can't withdraw from the trials and no human ethics approval is needed. Yet, the abstractness of our model allows us to take a broad perspective on interpreting interactions, and the small impact derived from student interactions with inappropriate objects can be viewed as the student refusing to work with the object or quickly giving up on the interaction.

For our simulations, the capability of delivering personalized learning experiences for each student is emphasized by running a variety of conditions for the students, the peer-base and the overall repository.

8.4.2 Other Applications of Simulation in ITS Research

Note that the reason we use simulated students is to assist in the validation of our algorithms for intelligent tutoring, towards the design of improved peer-based tutoring systems that employ a repository of learning objects. This is distinct from other uses of simulations for intelligent tutoring, such as work that has been done to allow students to learn in a simulated environment, ranging from flying a helicopter [68] to combat zone conversational skills [55] to surgical techniques [87]. That use of simulation for learning poses the additional challenge of trying to ensure that students are suitably engaged by the virtual environments that are presented. For our simulations, we only have virtual students and thus do not need to address this problem. We are able to abstract the details of the learning environment to simply represent the knowledge being taught.

Such simulations could be considered as learning objects, incorporated into our system and offered to students; these would be recommended by our curriculum sequencing approach in the same way as any other lesson.

8.4.3 In Contrast to Studies With Human Students

Our methods also contrast with those of others who conduct studies with actual human learners [37]. These researchers may be interested in examining the value of their approaches for much larger populations of students and thus the use of simulations may be of value. The robustness of the Collaborative Learning Algorithm to errors in assessment also provides encouragement for coping with inaccuracies in assessments which would undoubtedly occur when evaluating human students. Simulations allow us to observe the benefit of our approach in a environment with a very large number of students.

We note as well that simulations of learning are not a replacement for experiments with human students; however, the techniques explored in this work are useful for early development where trials with human students may not be feasible. While our current use of simulations is to validate our model, we may gain additional insights from the work of researchers [80] where simulations help to predict how humans will perform.

Our approach for simulated student learning can also be viewed as a specific proposal for representing student knowledge within user models that are employed within intelligent tutoring systems. Here, we offer a multi-dimensional distinction for each student (see Section 2.6.4.1), tracking knowledge levels in a set of different required major concepts. This aligns well with current user modeling efforts that promote multi-dimensionality in user modeling [86].

Perhaps most importantly, using a simulation to validate an ITS fills a much needed gap that is left when the value of these systems is confirmed through studies with human users alone. This is because: a) it is possible to simulate the learning achieved by a large number of students (whereas human studies are challenged to incorporate a very large sample size of participants) b) it possible to reflect the learning that would be achieved when a very large repository of objects has been experienced by previous peers (which is also a challenge when relying solely on the use of a human study).

In particular, our large corpus experiment (Figure 3.6) would have been challenging to assemble and provide to human students in a way that enables a wide range of the repository to be experienced. With a simulation, large corpus sizes can be created by changing a parameter. We do multiple runs with the 50 students that we inject into our simulation (for all 3 variants) which serves to be modeling an extensive learning experience, for each student. These experiments would be comparable to running an experiment for 50 students, in 20 iterations, for each of the 5 conditions: the equivalent of running a study with 5000 participants. With a simulated annealing approach, in particular, we are able to ensure that a wide range of objects are introduced to the simulation, exploring the full repository.

In addition, if our algorithm for intelligent tutoring were to be validated on the basis of a human study alone, we would be challenged in bootstrapping the system (assembling a record of peer experiences which form the basis of the future learning of the new students). We did, in fact, conduct a user study as an additional source of validation for the CLA (in Chapter 7). We ended up training the system with an initial set of 12 students and then used only this data to make recommendations for an additional 11 participants[2].

We also point out that validation of ITS has been an issue of some discussion, of late.

[2] We were unable to simply run the CLA as it is presented here, as this would have resulted in each participant having recommendations based on more data (the experience of all previous participants before him), which would have made for an inconsistent treatment group.

Looking at systems presented at ITS in 2010 (see Table 8.1) for instance, we note however that the number of people used in human studies was approximately 50 participants, which is considerable lower than we are able to achieve with our simulations.

Study	Participants
[50] and [89]	50 tutoring sessions
[41]	39 participants
[45]	25 students
[25]	6 course authors
[36]	4 classrooms
[24]	2 classrooms, 58 students
[56]	18 learners
[2]	106 students
[47]	105 students

Table 8.1: Representative User Studies from ITS-2010

8.5 Related Work

8.5.1 Intelligent Tutoring Systems

8.5.1.1 Personalized E-learning

In order to reflect on the value of our research, we begin with a comparison to related work conducted in the area of e-learning and intelligent tutoring systems.

Traditionally, intelligent tutoring systems researchers have been focused on providing opportunities for students to direct their own learning, making choices about lessons to explore [42, 13]. One significant way in which our e-learning framework differs is in its enhanced focus on determining the content to be presented to a student. In our approach, this content is determined on the basis of the experiences of other students and on the initiative of those students to introduce new subdivided learning objects into the corpus that is used as the lesson repository for the tutoring.

Moreover, compared to other intelligent tutoring systems, the content that drives the learning of each student is determined dynamically, through the possible creation of new divided objects, each time a student experiences learning with some existing object from the repository.

In addition, in our approach each student's learning is directed by taking into consideration all experiences of previous students, thus allowing for a continuous redirection of possible content. Personalization is maintained throughout, as well. This is achieved by modeling the knowledge levels of each student and an assessment of their current overall understanding in order to perform matching to like-minded peers, for the selection of learning objects.

Legaspi et al.'s work [49] has provided approaches to allow self-improving instructional planning. The authors created an agent that uses reinforcement learning to automatically derive categories and allows self-improvement of the system by using these categories to revise existing (or create new) instructional plans. Their approach requires extensive, explicit information about the learning objects (such as lesson goals and topic of instruction), students (such as cognitive ability, learning style, knowledge scope and lists of errors committed) and instructional plans (and, if they were ineffective, changes that could make them effective). Our approach, using data that will already be gathered in instructional contexts and not requiring creation of extensive metadata, can be viewed as faster, easier and cheaper to add to an existing system.

Also, in contrast to efforts such as Cheng and Vassileva [18], in our approach each student's learning is directed by considering all experiences of previous students, thus allowing for a continuous redirection of possible content. Personalization is maintained throughout, as well. This is achieved by modeling the knowledge levels of each student and an assessment of their current overall understanding in order to perform matching to like-minded peers, for the selection of learning objects.

8.5.1.2 Peer Based Tutoring

It is important to clarify that our use of the term peer-based intelligent tutoring is distinct from its standard usage within the AI and education communities. Typically peer-based or peer-tutoring refers to students learning by interacting with one another. Our approaches to tutoring can be viewed as peer-based, to the extent that student learning is enabled by previous peer interactions.

Previous work on collaborative learning [32] has attempted to use interactions between students and the system to provide a better experience for subsequent students (as we do). The authors created a program that would capture user problem solving behaviors in the system. This data was then used to begin the development of a tutor, in what they call “bootstrapping novice data (BND)”. The authors admit, however, that the task is non-trivial and reach the conclusion that that analysis must happen at multiple levels of abstraction. In contrast, our approach does not try to model specific user actions. Instead it pragmatically considers the sequence that learning material is experienced and how successful the students were.

COMTELLA The COMTELLA project [18, 84] at the University of Saskatchewan investigated recommendation of academic papers and to motivate the participation of users in a small-scale, on-line community. In the early phases of the work [84] there was difficulty in getting users to accurately provide metadata when entering papers in the system. Subsequent work [18] has focused on providing incentives to encourage users to interact positively with the system.

Motivations were primarily incentive mechanisms (marks in a computer science course) or visualization techniques. Vassileva et al. performed experiments on real students. One experiment was run with an upper-year computer science course where they were encouraged to share links with one another in an on-line community. Their system adapted itself to encourage users to provide what was needed by the community at different stages in the process (e.g. submission of links early on, ratings of links later on, meta-moderation after that) and to balance an active community with the danger of information overload (too many things being posted at once). They actively attempted to prevent the system from being “gamed”, that is, they endeavored to only reward students for the appropriate behavior and to prevent students from cheating the system.

They went further with this work by defining a number of metrics to measure a student’s participation in an on-line community. This includes such measurements as reputation, an adaptive expectation of participation based on the quality of past submissions, a comprehensive assessment of a user’s participation, and the weight of a user’s reputation that comes from sharing resources and evaluating resources shared by others. Statistical correlations between these metrics was determined.

Our work relies on students being motivated to participate in the system in order to leave annotations or to refine learning objects for subsequent students. Our content sequencing implicitly reasons about students using their standard activities, so there is no need to motivate them to take special actions.

In contrast we have the learning experiences of each new student directed by the previous experiences of their peers. This distinction is important because (i) we do not have to address the need to incentivize peers to advise their fellow students (we merely assume that the have agreed to have their past learning experiences anonymously analyzed and used within in the system) (ii) we have on hand a rich set of prior experiences that serve as the basis for directing the selection of content for new students and (iii) we are careful to avoid presenting potentially harmful content offered by peers.

iHelp iHelp[8, 83] is a project related to COMTELLA that involves reasoning about matching stakeholders (such as students, markers, tutorial assistants and instructors) in order to get the right information to the right person (both in public and private discussions). This typically involves real-time (or near real-time) interactions, where a question is asked and the system finds the right person to answer it. In later work [11] the authors extend iHelp to explore the value of tools such as chat rooms where learners are automatically drawn when using learning objects, shared workspaces where multiple learners can edit the same source code while discussing it and visualization tools for indicating a particular student's degree of interaction with her classmates. All of this is done to encourage "learner collaboration in and around the artefacts of learning". In contrast, our work seeks to provide repositories of useful information from past students, rather than provide tools to assist in the interactions between current students. In many cases these "past students" may be a classmate who used the learning object the day before, while in others it might be a former student who has since graduated and left the school. As such, our system has a greater focus on a long-term evolution where a large number of interactions are used to reason about how the system should present content to a student. Our focus also goes beyond matching individuals to decide which resources would be useful to present, to enable effective learning.

COPPER The COPPER system [66], explicitly arranges for students to practise conversations with one another for Intelligent Computer-Assisted Language Learning (ICALL). It intelligently matches students, and assigning them specific roles for their interaction, using Bayesian networks, "multidimensional stereotypes", and group modeling, and allows them to help one another learn through the creation of a framework integrating individual and collaborative learning to facilitate second language learning. Reed et al. organize students into groups where they would learn from their peers and assume a role appropriate to their aptitude level. An example might be two students who are assigned to work through a role-playing exercise of ordering food in a cafe, with a more advanced student assigned to evaluate their interaction. This is referred to in the work as "peer-scaffolding" and the claim is made that evaluating other students has pedagogical benefits to the student assigned to evaluation. Collaborations, such as these, are shown to be an effective method of learning a second language.

Their approach could easily be integrated with ours, where an interaction between students is a learning object. While their approach is useful in real-time, it doesn't allow students to independently learn from the experience of previous student interactions with the system. Our system, in contrast, reasons using the entire experiences of all previous students, not just the current, on-line students.

Our work also investigates structured ways for students to learn from their peers. While we do not focus on the creation of groups that work together on a problem, ways of thinking about similarities between users, such as "multidimensional stereotypes" and group modeling may be useful to explore in the future (see Section 9.2.2.5).

The Evolution of Social Relationships One valuable aspect of our approach in its management of the social network of peers is its ability to cope with a potentially large number of fellow students. This is achieved in part by first grounding the student learning in the context of a particular learning object that is most appropriate, based on the benefits in learning derived from this object by students at a similar level of knowledge.

Scaling is problematic for many approaches to real-time peer-tutoring [84]. Our framework, to respect the ecological approach, uses data from past interactions and performance improves as the size of the user base and repository of learning objects increases. A very large social network, therefore, is not a challenge at all, but instead an opportunity to provide highly personalized recommendations to students.

While computational demands do increase with a larger group of students, the time needed for such computations is small compared to the time it takes for students to complete tasks. The approach detailed in this thesis could easily scale to making recommendations for large numbers of students every 1/2 hour if needed[3].

Indeed, when a social network of peers is involved, a user does eventually need to make decisions about which peers to listen to. In the annotation part of our framework, we allow those peers who have not been helpful in the past to be redeemed and embraced anew within the social network (see our discussion of exploit vs. explore in Section 4.4.3); in addition, we do not blindly accept the advice that is offered by each respected peer, separately evaluating its worth towards the learning gains to be achieved. Our approach is similar to researchers who promote the modeling of peers in social networks for more effective decision making and learning by users [86]. In contrast with other intelligent

[3] We are encouraged by the speed with which our simulations, including some making over 5000 recommendations, were completed. If computation were to become a limiting factor, a straightforward adjustment would be to compute the predicted benefits for a student in between her interactions with the system instead of making recommendations on-the-fly.

tutoring systems researchers [66, 18], however, we are able to leverage past experiences rather than requiring peers to be assisting, in real-time with the learning that is achieved. As such, our users are free to browse and view the accompanying annotations on their web documents and videos, at their leisure.

Annotations The use of annotations in peer tutoring has been explored by a number of previous researchers. Read et al. [66] and the COMTELLA project [84] have investigated annotation techniques such as folksonomies and user tagging. While on the surface, this may seem similar to our work, there are important distinctions. With tagging, the purpose is to have users categorize items in ways that are meaningful for them, with the goal of sidestepping many of the problems inherent with ontologies (as articulated in McCalla's ecological approach paper [54]). In contrast, our approach endeavors to not just help students find an appropriate learning object, but to actually clarify that object and allow students to share insights with one another. Other work has been more explicit about arranging peer-tutoring [65, 66]. In their COPPER system, they arrange for students to practice conversations with one another, taking into account each student's level of proficiency, previous interactions and how they can best learn from one another. While our approach is a far less intense interaction than peer-tutoring that reasons about groups and gives them task in order to learn from one another, our approach has the benefit of allowing asynchronous learning. Students may be able to benefit from annotations left by students who are no longer even in the course. Work has been done considering text produced by learners, specifically the notes they take [84, 88, 52]. They used these notes and text retrieval techniques to implicitly derive information about the student and to build a profile and social network about them. In contrast, we take an intensely pragmatic view of annotations and don't try to decipher the meaning. Instead, our approach reasons directly about which annotation will help a student learn, and ignores the underlying content of the annotations.

The work of Lee et al. on the Vicarious Learner project [48], investigates how to automatically identify worthwhile dialogs to show to subsequent students, by determining the critical thinking ratio of a dialog, generated using a content analysis mark-up scheme. This ratio is determined from the positive and negative aspects within a discussion, with the assumption that discourse patterns provide signs of deeper levels of processing by learners and lead to a community of enquiry which benefits students. Dialogs with higher ratios could then be considered as valuable to show to new students. Our work differs from theirs in that we are interested in messages that have been explicitly left for future students and tied to a particular part of the course, rather than data-mining past interactions between students. Additionally, our approach is able to leverage similarities between students, in order to have a user-specific process for deciding which annotation should be shown. It may be interesting to integrate Lee et al.'s automated analysis of the critical thinking of text, as a component of deciding whether an annotation should be shown to a student.

We first note that there is value of being aware of possible bad advice from peers and avoiding it – not just for our context but for peer-based intelligent tutoring in general. The COMTELLA project [18] also deals with the situation of providing incentives to encourage student participation in learning communities. These incentives do not, however, eliminate scenarios where bad annotations may be left. Our work investigates this consideration. In addition, our approach does not focus on adjusting the contribution frequency of various students, but instead looks to preferentially recommend the more worthwhile contributions.

In Bateman et al.'s work on OATS [6], they explored annotations from the perspective of helping learners organize and navigate content. Much like this work, we are interested in leveraging the collective knowledge of groups of users rather than relying on manual annotation of a corpus by an expert. They divide social navigation support into "Traffic-based" (implicit) and "Annotation-based" (explicit); our work would be considered "Annotation-based". Their focus is on using annotations for "collaborative tagging", which refers to using social metadata where content is categorized based on an ad-hoc, user-determined categorization scheme. In contrast, we place no restrictions on annotations left by students and allow commentary on the actual content of learning objects. Additionally, while the OATS work reasons about similarity between users based on contributions to the system, we present a far more nuanced and sophisticated technique for reasoning about the reputability of annotations and annotators.

8.5.1.3 McCalla's Ecological Approach

McCalla's [54] work advocates leveraging past interactions with students in order to determine how best to interact with a particular student in a personalized manner. We go beyond this proposal, however, to allow for growth of the initial repository. This is important due to pervasive challenges in assembling appropriate tutorial content, as discussed by researchers [86, 58]. We therefore contrast with other researchers in peer-based intelligent tutoring [12, 18, 66] in allowing peers to (cautiously) play an additional role in the tutoring: that of content authors.

It is important to note that McCalla presents the concept of an ecological approach to ITS development in a far broader context than curriculum sequencing. He discusses applications of this philosophy to provide diagnostic advice to the student, to find a human helper (tutor or peer) or to help a student locate a learning community. However, there is no specific proposal in the ecological approach for enabling effective curriculum sequencing as considered in this work, and as well no use of simulation for validation.

8.5.1.4 Emotions and Learning

In the work of Graesser and Tanner [28] positive correlations were shown between students being confused and their post-test scores. Graesser et al.'s work is an interesting approach to measuring the impact of boredom, engagement, frustration, confusion, delight, surprise and neutral emotions on a student's ability to learn. They conducted their experiment by having 30 students work with an established ITS (AutoTutor) for 35 minutes with a pre-test and post-test to measure their learning. They found the pre-test and confusion to be the only significant predictors of post-test scores. The learners review video footage of themselves during the experiment and indicated their judgement of their affective state at that point in the instruction (previous work tried to do the same thing with emote-aloud or trained judges taking the place of self-reflection by the participants).

While on the surface our work may seem to over-focus on performance, in actuality this is not the case. Our system does evaluate the learning gains from a student's interaction with a learning object, but the specifics of this interaction are completely abstract. A learning object could be a 35 minute interaction with AutoTutor, and part of the learning gains made could be from the student experiencing confusion. If this is the case, our system will record the resulting improved learning and recommend that learning object in preference to objects without such effective pedagogical approaches. Our approach is agnostic towards the specific pedagogy and pragmatically focuses on the long-term learning of students and their similarities to other students.

We also note that one possible criticism of Graesser's work is that a correlation between confusion and learning does not indicate causation. One possible confound might be that students who care about learning the material are both more likely to learn and to be confused compared to students who wish to complete the experiment and are indifferent towards their learning performance.

8.5.2 Simulation

8.5.2.1 Simulated Students

We have introduced experiments that simulate student learning, in order to validate our proposed models and techniques for intelligent tutoring systems. Other intelligent tutoring systems researchers have previously explored the value of simulating students.

Van Lehn et al. [80] discuss experiences with simulated students and the methods that can be used to assist in education. The authors claim that this is useful not only for providing a collaborative learning partner for a student but also for instructional developers to test systems that they develop, including early development where trials with human students may not be feasible. This aim of simulated students coincides well with our motivation for using the technique, to enable initial validation of our proposed approach (and especially for contexts which might potentially involve very large numbers of users). The authors highlight grain-size as an important spectrum for considering simulated students. An example of fine-grained knowledge in physics is knowing the existence of tension in a string, when a string is tied to a body; simply knowing the law of conservation of energy. Our system uses a granularity outside of this range, which we would term coarse-grained. As an example, a student might be modeled as having a 0.67, which could mean, for instance, that the student has enough knowledge to receive a 67% mark in Physics 101 or that they understand enough knowledge to complete 67% of the projects.

Van Lehn et al. [80] also specifically track and formally represent, for each student, their behaviour during the learning and the specifics of instruction. In contrast, we are interested in tracking behaviour with respect to learning objects, and focus on modeling the student's knowledge before and after interactions with those learning objects.

Another research group used what they call learning curve analysis to analyze how their simulated student performed [53]. They measured the accuracy of production rules, in terms of successfully matching a step in solving the problem, compared to number of training problems or frequency of learning opportunities. We follow a similar approach in the evaluation of our work, where we use the resulting learning curves to contrast educational environments.

Matsuda et al. [53] also used simulated students as a technique for understanding the behaviour of real students. After training a simulated student using logs of interactions with real users, the simulated student they developed could explain 82% of correct problem solving steps performed by subsequent students.

Our system is different from these, in that our simulated students are used entirely to evaluate the efficacy of our techniques, and are not used as peers for humans or to predict their actions.

8.5.3 Educational Data Mining

Aligned with the aims of the educational data mining community, for our tutoring we are making use of data regarding previous interactions with learning objects from other students and the reactions of those students to the associated annotations. We are also modeling the learners in the community, first through our use of pre- and post-test assessments in order to have the learning of a new student influenced by that of previous, similar students and secondly through a tracking of a student's rating behaviour, in order to connect students more closely to annotations liked by similar raters. Most importantly, we are promoting data-driven adaptation and personalization, with our focus on reasoning about which annotations to present to a new student, over time, by virtue of what the data about previous experiences of similar students reveals.

While the data that is mined in educational data mining research is often that of real students (and we explore this option as possible future work in Section 9.2.6.3), there are a set of researchers more focused on making use of simulated students (as we do in our approach). For example, Rupp et al. [70] use principles from item response theory and diagnostic classification models in order to evaluate their approach to intelligent tutoring without necessitating real data collection with human students. In addition, Desmarais et al. [23] introduces four approaches to predict student data using simulations and then compares the performance of each to a student model trained using real world data. In contrast, our simulations are intended to model what real students may experience, but we develop independent measures of the effectiveness of our simulations (our approach of modeling the knowledge gained by students and the mapping of the mean average knowledge of the community of students). We have conducted a preliminary study as well (Chapter 7) to explore how well our particular simulations predict the performance of real human users, in order to fine tune our models.

8.5.4 Recommender Systems

Specifically within the ITS domain, there has been work done [46] on similarity matching in lifelong learners. In this work the experiences a learner has had are codified as strings. Pattern matching with these string identifies other learners with a similar background and makes recommendations based on their experiences. Our system uses an implicit approach rather than requiring a learner's experiences be codified.

Matching to similar users occurs based on life events, such as a specific degree at a certain university or working at a specific company. Their results are presented transparently in a user centric approach where users can investigate the "trails" of similar users. Matching was done by using string metrics where life events are encoded into a token based string which is used to reason about similarities between users. Our work is distinct from the above approach, however, in a number of ways. Obtaining a history, and accurately categorizing a user's life events, will be a time consuming process that may be difficult to convince users to undertake. In contrast, our approach to curriculum sequencing uses typical ITS interactions and does not elicit anything specific from the user. Our annotations approach uses elicit ratings, but will make recommendations with an incomplete set of ratings, whereas Lebeke et al.'s system [46] requires full information to function properly. In their system user histories must be continually updated, with the ongoing issue of out-of-date user profiles. The data used by our system can be easily gathered in real-time by usage of the system and will be as up-to-date as their last usage of a learning object. Finally, our modeling of students is on the basis of their knowledge levels as reflected in pre- and post-test assessments and as such reflects a more concrete representation than a cumulative code.

Wan et al.'s work [88] considers text produced by learners, specifically the notes they take. They used these notes and text retrieval techniques to implicitly derive information about the student and to build a profile and social network about them. In contrast, we take an intensely pragmatic view of annotations and do not try to decipher the meaning. Instead, our approach reasons directly about which annotation will help a student learn, and ignores their underlying meaning.

8.5.4.1 Collaborative Filtering Recommender Systems

On the surface, it might seem that recommendation techniques could be applied directly in an intelligent tutoring setting. However, whereas most recommender systems endeavour to obtain an increasingly specific understanding of a user, an intelligent tutoring system seeks both to understand a user and to enable change or growth. In addition, in contrast to positioning a user within a cluster of similar users, we would like to model a continually evolving community of peers who are operating at a similar level of knowledge.

Some of the cutting-edge areas of recommendation research are more relevant to us. The work of Herlocker et al. [33], which explores what not to recommend (i.e. instead of seeking highly relevant items from a set, removing irrelevant items) is relevant in our context, where peer-created learning objects or annotations which have been found to lack benefit for student learning may be worth removing from the repository. We discuss a possible future path with "garbage collection" in Section 9.2.2.1.

In our work, by first identifying appropriate learning objects based on student similarity and previous learning benefits obtained by peers, we are in tune with collaborative filtering recommendation [7]. Our consideration of the reputation of the annotator, however, introduces a novel method for overcoming cold start issues: annotations may initially be more likely to be shown, on this basis. Our integration of the similarity of raters with the new student then additionally assists in connecting students at the same level of appreciation. We are in essence recommending objects at each point in the student's educational progress. As students learn, they may be inclined to leave more or less generous ratings on the annotations that they see, and this ensures that each new student is appropriately informed, for their possible learning.

There has also been more general research on social networking and the streamlining of content to be shown to users within these networks [74]. This research is distinct, however, because it does not consider in detail the learning and the knowledge gains that are achieved by the user. Our approach, motivated by intelligent tutoring, does integrate that element.

8.5.4.2 Preferences in Interactive Systems

Our work also contrasts with that of other researchers who have explored the benefit of collaborative filtering in artificial intelligence. Some of these systems employ preference elicitation [61] in order to continuously refine the user models, with each new interaction. In contrast, with intelligent tutoring systems, the goal is to change the student and thus the student is a constantly moving target, posing a challenge for the user modeling.

We offer here a specific approach for peer-based tutoring that makes use of a rich interaction history to personalize delivery of content for users; this serves to assist students in focusing their attention on the most valuable material.

8.5.5 Trust Modeling

Our research serves to emphasize the potential value of trust modeling for educational applications (and not just for our particular environment of education based on the selection of learning objects that have brought benefit to similar peers, in the past). Our inspiration for the integration of reputation modeling is the work of Zhang and Cohen [94], which proposes a weighted combination of private and public reputation, when judging the trustworthiness of an agent. In our model, the reputation of the annotator provides the public reputation of the annotation, and the ratings provided by peers for the annotation are the private reputation. In this work, the trustworthiness of peers providing ratings of agents needs to be modeled as well. In our work, we consider instead the similarity of the peers and their previous rating behaviour, to influence whether we will accept their advice. Whereas work by Zhang and Cohen [94] is concerned with possible deception from agents, in our case this is instead an issue of reliability of ratings (i.e. peers with greatly differing educational experiences). The same general framework can still be leveraged and provides a novel element of trust modeling, as part of peer-based intelligent education. Our work, moreover, suggests a novel path for trust modeling research, to be more concerned with similarity of peers, due to their rating behaviour.

Zhang and Cohen's work on trust and reputation [93] was built around the idea of providing an incentive for buyers to accurately report marketplace experiences. Their approach did this by having sellers offer a discount to the more influential buyers, on the grounds that their positive interactions would have the greatest impact on the sellers' reputation. In turn, the buyers have an incentive to honestly report dealings in order to become more influential (and receive better deals from sellers).

The actual approach followed, the Bayes-Nash equilibrium, incorporates the cost of producing goods, the value of non-price features offered to the customer, and the distribution of profit based on various combinations of features. This approach also incorporates, on the buyer's side, thresholds of reputable or disreputable sellers based on the buyer's personal interactions, advice from other buyers (taking into account the trustworthiness of the adviser) and a public reputation. There is also a "forgetting rate" where newer ratings are considered more relevant than older ratings.

A simplified version of this model was incorporated in our work, as annotations and modifications to learning objects are made by peers. Evaluating these modifications based on previous interactions with that peer, or based on their reputation, was shown to be worthwhile. A simplified version is reasonable, since within a marketplace there is high motivation for deception (and a large cost of being cheated). Within an ITS there is far less reason for students to want to deceive other students.

8.5.5.1 Unfair Ratings

Zhang and Cohen's work on unfair ratings [91] deals with situations where a rating provided is unfairly high or low and how to create a robust system for dealing with these. The focus of this work is primarily on public reputations for unknown agents. They detail previous work that used a collaborative filtering style approach to remove high ratings from similar agents, but was unable to identify unfairly low ratings. They also detail "Iterated Filtering", an approach which uses a probabilistic model to identify the upper and lower bounds of statistically relevant ratings. This approach requires a significant majority of fair ratings to be useful. Other approaches detailed, including GM-GC, TRAVOS, Bayesian Network and RRSMAN each had some limitations, but on the whole could be a worthwhile basis for a trust model.

Other considerations in this work are: whether a trust model should incorporate agent preferences (and their similarity to other agents' preferences) when reasoning about ratings and whether an approach can deal with an agent whose behaviour changes.

In this work they provide a categorization for trust models based on Private vs. Public and Local vs. Global (see Section 2.5).

This approach for reasoning about how highly to weigh personal experiences versus the experiences of other was useful to our work. There is a strong focus on recent ratings in this approach which was less appropriate for our work (since the experiences of other students, even from some time ago, are highly useful for reasoning about the best way to present information to the current student). There may be value in incorporating a decay factor when assigning the initial reputation of annotations left (based on previous annotation's reputations and how long ago they were created), as an annotator's authorship ability may change.

Our work takes a different perspective on Local and Global than in Zhang and Cohen's collected work. Rather than personal experiences (Private) being contrasted with the experiences of other buyers (Public), we contrast experiences with a particular annotation (Private) to the overall experiences with the annotation's author (Public). This is a novel contribution that can be considered in the e-commerce domain. The analogous business model would be reasoning about particular products or services independently of the business. For example, if a buyer thinks Microsoft Office is excellent, Windows 7 is ok and Internet Explore is awful, she may have a neutral view of a new product from Microsoft (the company that makes each of these products), but be enthusiastic about a new release for Microsoft Office and pessimistic about a new release for Internet Explorer. This helps mitigate the cold start problem and allows a business to be evaluated in a richer manner: rather than being good or bad, reputations can provide more nuanced recommendations.

8.5.6 Personalization for E-Health

There is other work in the area of E-Health that has demonstrated the importance of personalized content delivery and of leveraging social networks as part of that learning [21, 22]. This work tends to focus on promoting healthier lifestyles by encouraging reflection and discussions within the family through the use of a collaborative platform. Our approach is aimed instead at allowing individuals to better understand their health concerns and make informed decisions. Camerini et al. [15] proposes personalized delivery of video to users to educate about self-care of fibromyalgia. This work confirms several elements in our approach, including video objects, and supporting personalized selection of objects from a corpus. Like us, their user study compared the value of their approach with one that was less personalized. One notable difference is that our tailoring is based on modeling peer-experiences.

8.5.7 Social Networking

Social networking has been a popular topic, with different interpretations, within sociology, online services and human-computer interaction. Our work has our own different take on this concept, while maintaining an awareness of these other perspectives.

Within sociology social networks are used to reason about connections between individuals, such as Mark Granovetter "Strength of Weak Ties" [29, 72] work which shows how sometimes a weaker social connection to someone outside your community can have a greater impact on political beliefs than a much stronger tie to a local community member. Online services, such as Facebook.com, allow users to "connect" to other users, then communicate with one another using messages, sharing web links or multimedia and status updates. Users are able to define the community of other users whom they interact with. Within human-computer interaction, there has been work done on "Crowdsourcing" [34], which refers to breaking up a larger task and distributing the workload to many users, each of whom do a small portion of the work. Examples include "tagging" where users are given tools to apply metadata to multimedia, which is then used to index the items and help other users retrieve them.

In our work, rather than any of these perspective on social networking, we view students from an educational perspective. Rather than friends, peers in the network are students who have a similar level of understanding and perhaps learning style for a particular student. In fact, as the student learns they should become less connected to peers with a lower level of understanding, until eventually they are removed from the students peer-group altogether, and more connected to more advanced students in their peers group (and have increasingly advanced students added to their peer group).

Our annotations and corpus approach work is more similar to the Human Computer Interaction (HCI) perspective. These approaches allow a crowdsourcing of the preparation of educational material, both with the creation of annotations and the division and recombination of learning objects.

Our research was motivated in part by the needs of caregivers and patients in the context of home healthcare (an application in focus for the hSITE project [63]). In these scenarios, users may form social networks in a somewhat random manner, deciding themselves which objects and commentary provided by peers to view, as part of their learning. Our research provides a more principled approach for connecting users with those peers who provide definitive educational benefit.

8.5.8 Annotations in HCI

Marshall's work [52] on annotations incorporates a summary of academic work on annotations within the HCI field, a taxonomy she creates based on these various perspectives which provides multiple dimensions for evaluating perspectives on annotations and a study she conducts using marked up textbooks at a university used-textbook store. Her research verifies our assumption that students are diverse in their creation and consumption of annotations and provides further justification for the focus on personalization in our annotations work.

Chapter 9

Conclusions

9.1 Contributions

In this thesis, we have presented five central contributions, as follows

- an algorithm for curriculum sequencing

  This approach for reasoning about the learning objects to present to each new student, based on the benefits derived by similar previous students, honours McCalla's ecological approach and leverages a history of interactions (modeled in terms of pre- and post-assessments).

- an algorithm for presenting annotations

  Here, we enrich the opportunities for participation from peers by supporting the attachment of annotations to learning objects and the rating of those annotations by subsequent students. We then limit the annotations that are shown by reasoning about the reputation of the annotation (in terms of the author's reputation including reasoning about the similarity of the raters and the active student). This constitutes integrating a specific trust modeling approach.

- an algorithm supporting the inclusion of divided objects in the corpus

  Our approach is to allow peers to also contribute to the growth of the corpus by proposing divisions of learning objects, which are then available to be shown to each new student, for the curriculum sequencing. Our solution continues to reason about which object provides the most benefit to each new student, avoiding those divisions proposed by peers which are not effective.

- a framework for simulating student learning, of use in the validation of the three central techniques listed above

  Here, we develop a detailed set of metrics for modeling the knowledge gains of students, mapping the mean average knowledge of each student in order to determine the value of a particular intelligent tutoring approach. Included is a modeling of impact and target level of instruction. We introduce as well three variations for our basic curriculum sequencing which vary according to how the algorithm builds up from the cold start case. We also include two benchmarks (reflecting maximally high and low performance). The framework that we develop offers promise for the validation of intelligent tutoring systems in general, for scenarios where a large corpus of objects should be experienced and therefore provides valuable assistance to ITS designers.

- preliminary validation of our approach with a user study

  While preliminary, we have begun to develop a framework to enable confirmation of our algorithms, through interactions with human users. To date, we have focused on exploring the potential value of our curriculum sequencing approach, compared to a less principled method of selecting content for users. This was achieved by adoping a Wizard of Oz style of experimentation, enacting an initial phase of training with users receiving random selection and then measuring the value of learning achieved by users following our proposed curriculum sequencing, through a series of assessments. Initial feedback on the promise of integrating annotations and corpus division was obtained through exit surveys. All of this was done for the valuable application area of home healthcare.

9.1.1 Design Decisions

Several key design decisions have served to enable the valuable contributions outlined above.

9.1.1.1 Inheriting History of Divided Object

Our work examined inheritance of the interaction history of the parent learning object after a new version of the learning object was created (see Section 5.4.2). This decision was made to avoid the cold start problem of a brand new learning object about which nothing is known. Clearly the new learning object will be different in some way from the parent; however, the perspective was taken that it would be more similar to its parent than to another random learning object. Our simulation showed that the system was able to differentiate between parent and child learning objects using the standard CLA.

9.1.1.2 Can Experience Same Learning Object

The question of allowing students to repeatedly experience the same learning object was a matter of much consideration and discussion. On one hand, students may get frustrated and bored with the same material being shown to them repeatedly. On the other hand, if the student hasn't fully appreciated the content of a learning object or has gained a stronger background which will allow him to more fully appreciate it, a second experience may be worthwhile.

For our simulations we allowed repeated interactions with the same learning object. For our human study we allowed students to only see a learning object once. Both options allowed students to learn and validated our approach.

9.1.1.3 Timed Lessons

Initially we made the simplifying assumption that all interactions with learning objects took the same length of time, and reasoned about a curriculum in terms of the number of learning objects experienced. We enhanced this simulation to allow learning objects of different lengths to be experienced. For each student we used the same time, which may be reasonable for some learning objects, such as videos, but unreasonable for others, such as reading a text. Our compromise (a set time of instruction for each learning object) gave us a deeper simulation while still being tractable.

9.1.1.4 Coping in the Face of Incorrect Assessments

In real world learning environments the possibility of inaccurate assessments are very possible. We used Gaussian noise to model this process, as it is a non-uniform distribution of randomness (much like assessments errors, we expect the final assessment to be closer to the accurate value than some other value).

9.1.1.5 Multi-Dimensional Model of Knowledge

We used a multi-dimensional model of knowledge, as described in Section 2.6.4.1. This allowed an abstract, robust model of student understanding that supported the variety of simulations run.

9.1.1.6 Greedy God / Random Baselines

Our approach used Greedy God and Random assignments as standard baselines to evaluate the performance of assignment of learning objects. One of the key benefits of our work is that the results are delivered in an automated manner, by typical student usage, instead of being hard coded by an expert instructional material developer.

9.1.1.7 Personalization and Authorship

For the corpus approach where students divide a learning object, we wanted to incorporate the idea that a student's understanding affects the division they choose. We did so by having a personalization, where a student's knowledge level alters the target level of instruction of the learning object. This could be thought of as a junior student selecting the easier elements of a learning object or a senior student selecting the more advanced elements. For authorship we modeled a student's ability to select the important parts of a learning object and used this to modify the impact after the object had been divided. Before authorship was applied, we scaled the impact of the learning object by the relative length of time. Personalization moved the target level of instruction 10% closer to the student's knowledge. Authorship was a real number, [0,1] that would determine the probability that the impact would be increased by 10% by the division, otherwise it would be decreased by 10%. These values were chosen such that the student division would impact future student experiences, but that the inherent value of the learning object would remain a large factor.

9.1.1.8 Impact and Target Level of Instruction

Our model for learning objects included an impact and target level of instruction. The impact was represented as, for the ideal student, how much would the learning object change their understanding of the knowledge. In our simulations we used a maximum impact of 5% of the possible range of knowledge. The target level of instruction followed the same range as the student knowledge described above (a real number [0,1]). These two elements made it possible for us to effectively model the knowledge gains provided by learning objects.

9.1.1.9 Simulation of Learning

We used Equation 3.6 (along with the student's model of knowledge and the learning object impact and target level of instruction described above) to determine student learning. This formula modelled the decreased benefit of assigning students to an inappropriate learning object and was compatible with our model of knowledge (see Section 2.6.4.1). The combinations of parameters included in this equation were effective in providing an abstract model of learning that incorporated variation in the students and learning objects.

9.2 Future Work

9.2.1 Design of a Repository for Peer-Based Tutoring

With our techniques in hand, ITS developers need only compile a repository of learning objects and a method of assessment. Our system will then deliver a tailored curriculum for specific groups of students who use the system. Below we outline various ways in which to offer assistance with the design of the repository.

9.2.1.1 Insights into the Design of Corpora

One direction for future research is to offer some insights into how to create the ideal set of learning objects to be presented to students. For example, it would be interesting to try to identify learning objects currently missing from an existing corpus which could be deployed for pedagogical benefit.

One approach to accomplish this would be to investigate a mature, ecological, intelligent tutoring system with an extensive interaction history between students and all learning objects in the repository. For every student the best predicted benefit would be calculated (this would determine, for each student, the best learning object to assign to them). The absolute values of predicted benefits could then be compared, and the lowest predicted benefit would identify students that are being under-served by the current repository of learning objects. By investigating at what level of understanding these students are currently, and their particular needs, new learning objects could be created for this portion of the curriculum (which would then, hopefully, provide a higher benefit in the future for similar students). This would allow the targetted creation of new content that precisely addresses an objectively determined, quantifiable need of the student population.

9.2.1.2 Real World Repository

Connected to the cold start problem, we could investigate techniques for using a repository of information, such as a collection of instructional videos or multiple recommended texts on a subject as a basis for automatically generating the core of an ITS. This would entail the automatic division of material and creation of a rough set of learning objects, which could then be refined by students using the corpus approach (Chapter 5). A real world system, with material created then shown to students, would be particularly worthwhile.

9.2.1.3 Addition Schedule of New Learning Objects

Another direction for extending our model and its validation is information about how the size of a student population affects the student learning that can be achieved, in peer-based tutoring environments. McCalla hypothesized that as the peer base grows, student learning can improve. Our experimental results (Figure 3.6) show that with larger repositories of learning objects there are larger learning gains. This also suggests a direction for future research in the design of intelligent tutoring systems that learn on the basis of peers, trying to measure and quantify the relative value of an increased population of peers or learning objects.

How to introduce new peers and learning objects is an interesting question not considered in this work. It is expected that new groups of students will arrive who should dramatically benefit from the past experiences of previous students. Similarly, in a real world environment it would be expected that learning objects would be added, changed and removed from a repository as the educational needs of the course of study was adjusted. Removing learning objects is supported by our current model, and changing learning objects could be handled (the student experiences would be based on the new version which would gradually adjust the recommendations). The best way to handle the addition of new learning objects is a more complex question. In our corpus work, we introduced derived learning objects by having them inherit their parent's interaction history and thus avoid the cold start problem. Brand new learning objects wouldn't have any such history to base recommendations on. One promising avenue on this approach would be to adjust the recommendation algorithm such that brand new learning objects are randomly introduced to diverse students until a baseline of interaction history is achieved, at which point the new learning objects would be recommended like any other object. An alternative would be to hard code some student qualities which would override the system and direct them to the new learning object for a period of time, after which point the override would be removed and the standard recommendations followed.

9.2.2 Curriculum Sequencing

9.2.2.1 Garbage Collection Algorithm

To date, we allow our repository of learning objects to grow with each new division proposed by a student and considered all learning objects in the repository to be assigned to students. A garbage collection algorithm could be used to consider the set of all learning objects in the corpus, reasoning about which learning objects should be retained and which add little instructional value to any of the students who may use the system. In its simplest form this can be considered a threshold of performance, below which a learning object is no longer shown to students.

In this work, garbage collection was not performed. The CLA predicts potential benefit for every learning object for every student. For a learning object with a low predicted benefit for each student (a prime candidate for garbage collection) the CLA would be very unlikely to recommend this to a student, and therefore the same result of removal is achieved.

In future work, proceeding with the assumption that resource constraints (such as storage space for the learning objects and computational demands for reasoning about assigning low-quality learning objects) may provide benefit to restricting the number of learning object, we may examine explicit garbage collection of learning objects and the benefits it provides. Methods for determining "what not to recommend" [33] may be useful for this work.

9.2.2.2 Which Combination Makes Sense

Our curriculum sequencing approach is a greedy algorithm. For a student at a particular point in their course of study we recommend the learning object which is best predicted to improve their knowledge. Beyond combining two or more learning objects into one larger object (discussed later in Section 9.2.4.3), it may be useful to reason about sequences of learning objects that may reinforce one another and give a better experience as a set, rather than selecting individual learning objects.

This is a far more computationally demanding approach than what is presented in this thesis. Given a repository of N learning objects, a sequence of M interactions would require $\binom{N}{M}$ times as many calculations (so for a repository of 100 learning objects and reasoning about a sequence of 5 interactions would require 75 287 times the computational resources compared to our greedy approach). No doubt there would be the possibility for improvements over this worst-case.

9.2.2.3 Repeated Exposure to Learning Objects

It can be debated what the pedagogical value is of repeated exposure to the same learning object. From an intuitive perspective, it could be argued that students don't need to learn the same thing twice and that all material should be novel. Malcolm Gladwell [26] summarizes an approach followed by "Blue's Clues" (a children's educational program) that made the unorthodox decision to show the exact same episode 5 days in a row. The producers found that attention and comprehension would actually increase with each viewing. At a more advanced level, most graduate students have had the experience of reading a foundational paper in a new area they are trying to get up to speed in which helps them get an overview of the field and big concepts in it. Later, after reading more work in the area, reviewing the foundational first paper again will provide more detailed, nuanced points that they didn't appreciate in their first reading.

Psychologists such as Mace [51], Spitzer [76] and Pimsleur [62] have specifically investigated the learning technique of subsequent review of previously learned material.

In our simulations we took the perspective of a course of study where allowing repetition of learning objects was worthwhile. In our human study, due to the short length of instruction (approximately 1 hour) the decision was made not to repeat learning objects. Our current perspective is that either approach may be appropriate for a particular learning domain and student population. A hybrid approach, where repetition is allowed but heavily penalized when making recommendations, could also be considered. Evaluating these alternatives could form the basis for future work.

9.2.2.4 Similarity-Based Clustering

Work has been done by Qin et al. [64] on using clustering for preference elicitation. The motivation of this work was to minimize the number of required utility queries given to users (minimize the bother cost) while maintaining an accurate understanding of the user's preference. Their approach to this is to cluster similar users, then use the aggregate preferences of the cluster instead of individual users and spread the cost of queries across the group. They use a simple approach to estimate unknown utility for an individual: conditional outcome preference networks (COP-networks). First they create a directed graph representing strictly ordered preferences over a set of outcomes. This graph is created from the preferences of the cluster. For an individual user, once two preferences on the graph are elicited, unknown preferences are predicted by scaling these two values through the rest of the graph.

In order to cluster users, they performed Y-means clustering. Y-means is an extension of K-means clustering, which clusters items by minimizing the within cluster sum of squares. In contrast to K-means' fixed number of clusters, Y-means can dynamically adjust the number of clusters (by merging or splitting clusters).

Another interesting element of this approach is the directed graph can be used for error-correction of preferences. Any time that a cycle exists in the graph, it indicates inconsistencies in the reported utilities. By reversing the minimum number of edges in order to remove cycles, these errors can be corrected.

To date, our work has used approaches that precisely measure the similarity between two students. The margin of error in these measurements is open to debate, and an alternative would be to cluster users (to have create a rough approximation of "similar" users) instead of trying to produce fine-grained assessments of their similarity. This work provides a clear overview of one possible clustering approach.

9.2.2.5 Lifelong Learner Modeling

Judy Kay's work on Lifelong Learning Modeling [42] considers how to approach building a user model to support universal, personalized lifelong learning. One of the Grand Challenge Problems by the Computing Research Association and two of the nine Grand Challenges for Computing from the United Kingdom Computing Research Committee focuses on lifelong learning and modeling students. Kay advocates giving learners control of their model (what goes in, what comes out and the ability to examine the model itself). She also advocates the usage of ontologies as a method for standardizing services and allowing a learner model to exchange data with the multitude of software services they may encounter over the course of their lives and learning. McCalla [54] cast serious doubt on the ability of ontologies to provide this functionality.

Kay discusses stereotypes and communities as methods of reasoning about learners. An example of a stereotype is in the Unix Consultant [20] (which allowed, through the use of a single question, the ability to determine extensive information about a user's knowledge of Unix). Communities are like stereotypes, but allow partial membership (so a learner might be a 20% match with male stereotypes and an 80% match with female stereotypes). These can be used to reason about a learner within the model.

Kay's vision of learner modeling also includes the ability to have the model spread over various devices and applications. In order to provide scrutable user models (models that the user can understand and reflect on), she advocates a focus on interfaces and discusses techniques such as those for groupwork visualization.

User models develop an increasingly detailed understanding of the user with each interaction. In contrast, with intelligent tutoring systems, the goal is to change the student and thus the student is a constant moving target, posing a challenge for user modeling. In our work we sidestepped this challenge by considering all student models as distinct and separate students after every interaction with the system. To be sure, it can be expected that they will be very similar to their "past selves"; however, in the domain of content sequencing in particular it can be counter-productive to cling to an understanding of the student which is (hopefully) being rendered obsolete through interactions with the system. While we observe that there is immediate benefit in treating each student interaction as a separate student, we also acknowledge that some techniques to model the trajectory of student progress may be useful for future work.

Learner modeling will certainly be a central element of future research stemming from this thesis, and stereotypes and communities might be a sophisticated way to create simulated students who approach an intelligent tutoring system in different ways. As we proceed with additional experiments with humans, the design of the interface will be highly important and Kay's ideas may be useful to guide our approach to this part of the experiment.

In Section 5.4.5 we discussed modeling a student's proclivity for shorter or longer learning objects. Running simulations with additional variants on student learning may provide for an even more personalized algorithm for curriculum sequencing and insights from Kay's work may be useful for this approach.

9.2.3 Annotations

9.2.3.1 Deeper Student Models

As mentioned previously, for annotations we simulated students as accurately rating (a thumbs up or thumbs down) annotations based on whether the annotation had helped them learn. It would be interesting to provide for a richer student modeling where each student has a certain degree of "insight", leading to a greater or lesser ability to rate annotations. If this were incorporated, each student might then elect to model the rating ability of peers and this can then be an influence in deciding whether a particular annotation should be shown. It might also be useful to model additional student characteristics such as learning style, educational background, affect, motivation, language, etc. The similarity calculation would need to be updated for such enhancements; similarity should then ideally be modeled as a multi-dimensional measure where an appropriate weighting of factors would need to be considered. Similarity measures such as Pearson coefficients or cosine similarity may be appropriate for this approach.

9.2.3.2 Trust Modeling

Future work could consider integrating additional variations of Zhang's original model [94] within our overall framework. For example, we could start to flexibly adjust the weight of Local and Global reputation incorporated in the reasoning about which annotation to show to a student, using methods which learn, over time, an appropriate weighting [94] based on when sufficient Local information is available and can be valued more highly. In addition, while trust modeling would typically have each user reasoning about the reliability of every other user in providing information, we could have each student maintain a local view of every other student's skill in annotation (though this is somewhat more challenging for educational applications where a student might learn and then improve their skill over time and where students may leave good annotations at times, despite occasionally leaving poor ones as well). In general, studying the appropriate role of the Global reputation of annotations, especially in quite heterogeneous environments, presents interesting avenues for future research (since currently this value is not, in fact, personalized for different users).

We plan to further explore trust-based modeling of peers [94] in intelligent tutoring environments; to this end, the work of Gorner [27] which uses trust modeling to investigate how to determine the ideal size of social network for providing appropriate advice may also be of some interest. In contrast, other trust modeling researchers have proposed the limiting of social networks of advisors, in order to make the trust modeling process more efficient and effective, when used to drive the decision making of a user [77, 90]. Our algorithms do incorporate a process of restricting the set of advice that is considered; but beyond this we also integrate the elements of similarity and modeling the tutorial value of each new object that is presented.

The Chernoff Bound Theorem [19] they used may be useful for our annotation work to determine the minimum number of rating pairs in order to provide an acceptable balance of level of error and confidence in measurement when reasoning about a given public and private reputation. In their work they used a simple simulation which was somewhat similar to our curriculum sequencing work. This reinforces the appropriateness of our use of simulation as a validation technique.

Credibility In their works [73, 72], Seth et al. investigated trust and credibility within the context of weblog messages and other participatory media. Their approach used Bayesian learning, based on multiple sources of information to establish credibility. These include personal experiences with the author, the author's role (student, journalist, etc), the reputation of the community the author is a part of, and the public opinion on other messages the author has left. After dividing data into a training and a testing set, they evaluated their approach using Matthew's correlation coefficient (MCC) and TPR-FPR as metrics.

Elements of their work would be useful extensions to the trust model in our work. In particular, one concern is the development of folklore where student populations may convince one another of something that is not true, then reinforce participants who agree with the misconception and punish those who try to correct it. A role-based element to the trust system would allow participants with acknowledged expertise, such as the instructor or teaching assistants, to have a disproportionate impact when they rate content. This would allow experts to correct misinterpretations that the student community had fallen into.

9.2.3.3 Student Ratings of Annotations

In our work we made the simplifying assumption that students could accurately report the benefit they had received from annotations and provide perfectly accurate ratings. We were confident, with the result we had with errors in the curriculum sequencing (see Section 3.6.1), that the annotations approach should be robust to errors as well. This is an assumption that could be relaxed in future work in order to determine the robustness of the model.

9.2.3.4 Personalized Annotator Reputation

Much the same way that we personalize individual annotations (see Algorithm 4), we are interested in also personalizing the annotator reputation. This would allow reasoning about highly reputable or disreputable annotators for specific subgroups in the population of students, which may be different from the view of that annotator by the entire student population.

9.2.3.5 Allowing For Errors in Judgement

Our simulation modeled insightful students who are able to accurately rate whether or not a specific annotation has helped them learn. In practice, the ability of students to make this identification may not be as accurate as we have assumed in this work. In future work we are interested in exploring the impact of modeling some uncertainty in this process, whether student sometimes give an incorrect rating to annotations they have experienced.

We are also interested in examining this issue with real human students. In both cases this would be an interested experiment to re-examine in those contexts.

9.2.3.6 Gaming and Collusion

As with any educational environment, gaming and collusion are possibilities. Gaming occurs when a student is able to improve their assessment without learning the material. Collusion occurs when a group of students collaborate in order to cause a certain outcome to occur. Baker has done extensive work [5] on detecting and dealing with gaming, which could be applied to this approach if this were an important consideration for a system under development. In environments such as home healthcare, users may be quite inclined to help each other to learn, and thus collusion should not be prevalent.

9.2.3.7 New Kinds of Annotations

Our view of annotatations could certainly be broadened to include functionality such as that provided by iHelp [11] where a chat room is attached to a learning object. From the current perspective of this work, we would either consider the chat room to be a learning object with its own benefit that might be assigned to a student, or would allow a text link to the chat room to be used as an annotation on a learning object.

9.2.4 Corpus Division

9.2.4.1 Motivations from Text Processing

Work on tailored summaries [60] attempts to concisely capture the important ideas of a document, taking into account the user and the context the document is accessed in. This work investigates these ideas through the use of an in-browser text summarization tool. The primary intention was to maintain the user's focus, by helping him avoid being side-tracked by tangential links and to remind him of what he was looking for when he accessed a web page. When the user moves his cursor over a link, a customized summary of the linked-to page is shown, using the current page and the text of the link to customize the summary.

Our current approach uses a vector spaces approach with the Manhattan distance used to measure the similarity. Their approach also uses a vector space approach, but with a cosine similarity metric instead. It operates at the sentence level (with the linking text taking the place of keywords), identifying sentences in the target document that match the general user interest, as represented by the original document, and the specific user interest, as represented by the linking sentence. This is simple, scalable and doesn't require sophisticated natural language processing techniques.

Our current approach to refining learning objects is to enlist students to highlight the most valuable parts of a divisible object. Their approach could be modified to be used as an alternative for text-based learning objects. That is, we could use their approach to automatically generate streamlined, targeted learning objects like we are doing in this work and contrast the two approaches. The most relevant parts of the learning object could be automatically extracted in a customized manner from a repository of objects.

Another alternative is to create a hybrid, mixed-initiative version of the two approaches, where tailored text summarizes could be used to help students to streamline text learning objects.

9.2.4.2 Tracking Student Corpus Division Habits

Wan et al. [88] have done work on considering text produced by learners, specifically the notes they take. They used these notes and text retrieval techniques to implicitly derive information about the student and to build a profile and social network about them. This has similarities to our annotation work, as both attempt to leverage material generated by students to assist other students in learning. For future work, there may be value in modeling the corpus division habits of a particular user, in order to represent that in the student's user model, of use in connecting to like-minded students for the peer tutoring.

9.2.4.3 Combining Learning Objects

A further potential extension is the idea of combining learning objects which are commonly assigned to the same students at the same point in their education, or combining learning objects that students feel illuminate one another. For a curriculum sequencing approach, learning objects which were often assigned after one another could be merged in a manner similar to that described in Chapter 5 (that is, the system would implicitly reason about which learning object would benefit students). The new learning object would consist of the two learning objects, presented together or one following the other. The interaction histories of both learning objects would be attached to the new learning object (and a differentiation between the combined object and the two original objects could be made in a similar manner as described in this work). This would, over time, tend to assign the combined object to students who would benefit from it, and the individual objects to students who would not.

9.2.5 Simulation

9.2.5.1 Additional Experimentation

Various new directions for adjusting the modeling within our simulation are useful to investigate. These include:

- altering the exposure of the students to the repository of learning objects: in other words, investigating the impact on system calibration when exposing all students to the entire system vs. some students to the entire system, vs. some students to part of the system, vs. all students to part of the system vs. known students to the entire system. Perhaps if exposing students to part of the system, we would investigate different learning "growth rates" of incorporating the rest of the system into the "active lessons". This may provide further insights into how best to handle the cold start problem.
- extensive experimentation to discover the ideal choice of the ecological approach (raw ecological versus ecological with pilot versus simulated annealing); we suspect that the ideal choice will be domain specific.
- modeling student authorship skill more precisely by tracking the learning achieved by students when interacting with objects authored by a particular peer.
- modeling learning objects at a deeper conceptual level; divisions could then be made on the basis of which concepts were being taught within a specific part of the larger learning object.
- investigating impact being sensitive to whether students are strong or weak learners (so incorporating more student modeling).

It would be valuable to explore varying several of the parameters in our current simulations. These include:

- experimenting with exponents other than 2 in the denominator of Equation 3.6; this would allow for imposing greater or lesser penalties for being assigned a learning object with an inappropriate target level of instruction[1]. We suspect that the best value for this exponent will be domain dependent.

- exploring metrics other than the mean average knowledge; there may be some value in clustering students into subpopulations and tracking separately the knowledge gains of each or in identifying and removing outliers.
- providing weights for the various dimensions of knowledge such that they do not all contribute equally to the student's final assessment.
- incorporating a requirement that students must complete a variety of learning objects of varying lengths. However, it might be challenging to provide each student with a comparable learning experience. An alternative to forcing a variety of lesson lengths would be to assess students continuously as they experience learning objects.
- exploring the use of other distance metrics when calculating student similarity for multi-dimensional knowledge.
- allowing students to have a non-static authoring ability for annotations and corpus-division. This would enable us to model improvements that students achieve when progressing through the curriculum.
- varying the 20% division parameter for student division of a learning object; this would enable the modeling of students who are more or less willing to suggest divisions.
- experimenting with divisions in our simulations of the corpus work that are fractions other than 1/2; this may make it somewhat more challenging to track the performance of our algorithm for reasoning about which objects to show to students, especially if there is great variability in the length of objects which are generated by the students proposing the divisions.

9.2.5.2 Particular Challenges of Modeling Student Learning

Our work also contrasts with that of other researchers who have explored the benefit of collaborative filtering in artificial intelligence. Some of these systems employ preference elicitation [61] in order to continuously refine the user models, with each new interaction.

[1] This value should be an even number, otherwise there will be an asymmetrical effect on the learning depending on whether the student's knowledge is greater than or lesser than the target level of instruction.

In contrast, with intelligent tutoring systems, the goal is to change the student and thus the student is a constantly moving target, posing a challenge for user modeling.

In our work we sidestepped this challenge by considering all student models as distinct and separate students after every interaction with the system. To be sure, it can be expected that they will be very similar to their "past selves"; however, in the domain of content sequencing in particular it can be counter-productive to cling to an understanding of the student which is (hopefully) being rendered obsolete through interactions with the system. While we observe that there is immediate benefit in treating each student interaction as a separate student, we also acknowledge that some techniques to model the trajectory of student progress may be useful for future work.

With respect to our modeling of knowledge and simulation of learning, we note the consequence of this approach includes a symmetrical degradation of the usefulness of the lesson (whether student is too advanced or not advanced enough, the lesson's impact decreases in the same way). This seems to be a reasonable simplification; however, it may be worthwhile to consider other models of learning where inappropriate assignment of learning objects provides different results. Some different models for learning [59, 79], could be considered when simulating the interaction between learning objects and students. This would allow both our simulation approach contrast various learning theories, and to provide an initial assessment of the impact of an ecological approach to systems built based on these descriptions of learning.

9.2.6 Additional Validation

9.2.6.1 Additional Human Studies

For additional human studies we could consider:

- allowing divisions proposed by participants to be included in the repository that is used to teach future students.

- experimenting using different domains for the teaching, to see if this would result in differences.

- ultimately moving away from the Wizard of Oz approach and designing a fully automated system with an effective user interface.

- incorporating our techniques into various learning environments such as the Tactical Iraqi Culture Training System [40] both to demonstrate their effectiveness in diverse contexts.

9.2.6.2 Model Evaluation

In Jastrzembski and Gluck's work on comparison of model variants [38] the authors raise the concern that simply comparing the "goodness-of-fit" between a model and real-world data is insufficient justification of the validity of that model. They explore 3 additional criteria which they feel should be considered and present 3 models of learning, real world pilot learning data and contrast the models using these their proposed criteria.

Their Bayesian Information Criterion (BIC) evaluates models based on their ability to predict future data samples, penalized by number of parameters and weighted against goodness-of-fit. Cross-Validation breaks the available data into two subsets, the first is used for parameter calibration and the second is used to evaluate the models predictions. Minimum Description Length (MDL) is simply a measure of how complex a model is, with the idea that a simpler model is better than a complex model (all other things being equal). This paper provides 3 extensions beyond simply comparing the datasets, each of which can be used to illuminate similarities and differences in simulated learning compared to real world learning. This might affect how our simulations would be performed in the future.

9.2.6.3 Post-Hoc Data Analysis of Real World Human Data

Another useful starting point for additional validation would be to begin with extensive data that represents educational paths already completed by students and to determine whether our algorithm would have made effective choices for these students. In particular, if we had data that tracked student course selection and grades throughout their undergraduate career, if those paths that led to higher overall grades matched well to our proposed curriculum sequencing this would be an additional validation of our approach (and conversely, if those paths that led to lower overall grades matched poorly with our proposed curriculum). Post-hoc analysis refers to obtaining data not explicitly from an experiment, but instead from a real world source which is evaluated after the fact.

We could imagine carrying out this procedure as follows. Consider a data set consisting of the course grade history for all students in a department at an institution. Starting chronologically at the beginning of the student data, we would step through term-by-term and see what the CLA would recommend for students about what courses they should take in the next term. The interaction history which informs the CLA would be all courses taken by other students in previous terms. For the first term no recommendations would be made, since this is the cold start problem. Benefit would be modeled by comparing the student's average mark in one term as their pre-test and their average mark in the next term as their post-test. Similarity would be determined, again using a multi-dimensional model of knowledge, where each course is a dimension, and the marks are the values that are compared. For example, two students who had two courses in common and had received marks of (80, 89) against (60, 69) would be far more similar than two students who had only one course in common and marks of (84) against (74) and 4 others that were not in common; this is because the courses not in common would be considered to be maximally different.

Each term the recommendations made would be compared to the courses actually taken by the student and a metric calculated, measuring how closely the student followed what would have been our recommendation (e.g. if he took 4 of the 5 courses that we recommended, we would record that he followed 0.8 of our recommendations). The student's actual average for the term would also recorded. After these values were determined, the correlation between them would be computed (using linear regression), with the expectation that students who followed what our recommendations would have been would outperform students who did not.

Some of the challenges we would face include: course codes changing over time, new courses being added and lacking interaction history, courses being removed, matching students who may take a term off or take a co-op term and not be enrolled in classes (One way to overcome this later challenge is to consider only the term when a student was enrolled in courses, and ignore terms where they worked or where they were not at school.). Another decision to make is whether to ensure that students who are being compared are at the same term of study; not imposing this requirement might fail to completely capture similarity but would have the advantage of more robust matching of student learning.

For future work we would explore obtaining data such as this from, for instance, the Cheriton School of Computer Science at the University of Waterloo. We suspect that this kind of data would present additional challenges due to its extensive size (for memory and efficiency of processing).

Students often take courses on a variety of subjects. For example, a first year student in Mathematics at the University of Waterloo might take the following courses:

- MATH 135: Algebra for Honours Mathematics
- MATH 136: Linear Algebra 1 for Honours Mathematics
- MATH 137: Calculus 1 for Honours Mathematics
- MATH 138: Calculus 2 for Honours Mathematics
- CS 135: Designing Functional Programs

- CS 136: Elementary Algorithm Design and Data Abstraction
- PHYS 121: Mechanics and Waves 1
- PHYS 122: Mechanics and Waves 2
- ECON 101: Introduction to Microeconomics
- PSYCH 101: Introductory Psychology

Recommending courses raises a new challenge. Should only mathematics courses be recommended or perhaps all required courses, that is Mathematics, Physics and Computer Science? Should electives be recommended as well? Electives often give higher marks than core courses, so allowing electives in the recommendation might lead to the curriculum sequencing approach recommending full course loads of electives in order to maximize students' expected marks. Imposing course selection requirements would be an onerous processes, especially considering that these change over time and can be overridden by advisors and therefore may be difficult to capture from student records. It may make sense to begin by focusing only on core courses (the mathematics courses in this case).

9.3 Summary

In this work we have presented three concrete approaches for creation of a personalized, adaptive intelligent tutoring system for environments that make use of McCalla's ecological approach to intelligent tutoring – utilizing other students' previous interactions to determine the best way to interact with a current student. In addition to these models, a novel approach to simulating learning environments was developed and used for validation. A human study provided further validation and informed our approach. These approaches are amenable to real-world deployment and provide an efficient way to enhance the performance of an e-learning system while decreasing the development costs.

This work provides a rich array of paths forward to further explore the issues examined. Educational recommenders, peer-based learning guidance and social navigation are all hot topics for a number of research communities, and the approaches outlined in this work has the potential to make significant contributions in each of these areas. For future work, we plan to extend our models and examine the impact of adding additional parameters that allow modeling of a greater variety of students and learning domains. We intend to conduct additional user studies, with a goal of further validating the work presented and to inform progress on these models. In fact, we are currently in talks with educational software companies who have expressed interest in our work and hope to assist them in integrating these approaches with their products and evaluating the impact on the students using them. Finally, we have obtained student data from computer science and mathematics students at the University of Waterloo: we plan to apply our techniques, in a post-hoc manner, on this data to examine the impact recommendations could have made if provided to this student population.

In all, we view our research as offering novel, well-validated techniques of value to the construction of peer-based intelligent tutoring systems.

APPENDICES

Appendix A

Assessment Questions from Study

An asterix is next to the correct answer.

Participant #: Assessment # (1-6):

1. How likely is an autistic child to go to college / university?
a. More likely than non-autistic children
b. Equally likely as non-autistic children
c. Teenagers with autism are unable to go to college or university
*d. Less likely than non-autistic children

2. Which of these is the specific cause of autism?
a. Immunization
b. Cold parenting (e.g. lack of cuddling and affection)
*c. The cause is unknown
d. Alcohol consumption during pregnancy
e. Genetics

3. What of the following is NOT usually recommended as an early next step after a child is diagnosed with autism?
a. Joining a support group
*b. Switch child to a gluten-free diet
c. Begin assembling an intensive treatment program
d. Ask for help from friends and family
e. Participate in Autism Spectrum Disorder research

4. How likely is a boy, compared to a girl, to be diagnosed with Autism?
a. Half as likely
*b. Twice as likely
c. Equally likely
d. Eight times as likely
e. Four times as likely

5. How is Asperger's Syndrome different from autism?
a. Asperger's Syndrome is a more serious form of autism
*b. Individuals with Asperger's Syndrome lack the speech delay that children with autism have
c. Asperger's Syndrome is the male version of autism
d. Individuals with Asperger's Syndrome, also known as savantism, have one or more areas of expertise, ability or brilliance that are in contrast with the individual's overall limitations
e. Asperger's Syndrome and autism are alternative names for the same disorder

6. What would be the most salient reason for a temper tantrum in a child with autism that differs from a child without autism?
*a. They are caused by difficulty communicating
b. Parents don't know how to discipline children with autism
c. They correspond to a poor diet from finicky eating
d. Are unavoidable and nothing can be done about them

7. Which of the following is true about sensory issues accompanying autism?
a. Often disappear as child ages
b. Are a specific expression of poor behaviour
*c. Vary greatly in their expression and severity
d. Are usually caused by underlying physiological issues
e. Are typically visual issues

8. Which of the following is NOT a possible sign of developmental delay in a child that should be tested for autism?
a. No single words by 16 months
*b. By 24 months non-family members can't understand the child's speech
c. Any loss of any language or social skills at any age
d. No 2-word spontaneous phrases by 24 months, with the exception of repeated phrases (echolalia)
e. No babbling, pointing, or other gestures by 12 months

9. Which of the follow is NOT a standard therapy for children with autism?
a. social skills training
b. speech therapy
c. physical therapy
d. occupational therapy
*e. gestalt therapy

10. Which of the following is NOT a physical or medical issue that commonly accompanies autism?
a. Sleep Dysfunction
b. Gastrointestinal Disorders
c. Sensory Integration Dysfunction
*d. Dermatitis or Eczema
e. Seizure Disorders

Appendix B

Survey About Learning Object

Streamlining of Learning Objects

Participant #:
Learning object order (e.g. 2nd)
Learning object # (i.e. 1-20):

1. How would you rate your satisfaction with the learning object you saw?

-5	-4	-3	-2	-1	0	1	2	3	4	5
unsatisfied					neutral					satisfied

2. How would you rate the appropriateness of the learning object for what you want to learn about caring for a child with Autism Spectrum Disorder?

-5	-4	-3	-2	-1	0	1	2	3	4	5
inappropriate					neutral					appropriate

3. How would you rate the appropriateness of the learning object, given your background knowledge about Autism Spectrum Disorder (i.e. Was the learning object at the right level or was it too basic or too advanced)?

-5	-4	-3	-2	-1	0	1	2	3	4	5
inappropriate					neutral					appropriate

4. How did you find the order in which the learning object was presented?

-5	-4	-3	-2	-1	0	1	2	3	4	5
unsatisfied					neutral					satisfied

As has been verbally explained to you by the researcher, part of our work is investigating the idea of streamlining learning objects (such as magazine articles, YouTube-style video and books). To this end, we are investigating having participants highlight the parts of a learning object they found most useful in order to provide a more targeted learning object to subsequent, similar participants.

Imagine you had a tool that would allow you to do this:

1) Might you create a streamlined version of this learning object? Yes / No

2) If yes, on the learning object you just worked through, please highlight (the researcher with you will provide a highlighter) the sections you would KEEP in the streamlined learning object. If it is a video, please list the segment (timestamp to timestamp) you would KEEP below. Feel free to highlight multiple sections and ask if you have any questions.

If no, why not?

Appendix C

Exit Interview from Study

Project Title: Peer-Based, Personalized Curriculum Sequencing for Information about the Care of Children with Autism Spectrum Disorder
Project Investigators: John Champaign, PhD Candidate & Robin Cohen, Faculty Supervisor
David R. Cheriton School of Computer Science, University of Waterloo

Exit Survey for Participants

Participant #:

Date and Time:

For each of the questions below, circle just ONE response. Choose the one that you feel is most appropriate.

I With regard to the streamlining of learning objects you were asked about after each learning object.

1. How would you rate the difficulty of creating a new streamlined learning object?

-5	-4	-3	-2	-1	0	1	2	3	4	5
difficult					neutral					easy

2. How would you rate the difficulty of deciding what content to include in a streamlined version?

-5	-4	-3	-2	-1	0	1	2	3	4	5
difficult					neutral					easy

3. How would you rate the usefulness of a system offering a user the full version or streamlined version of content like you've seen?

-5	-4	-3	-2	-1	0	1	2	3	4	5
useless					neutral					useful

Any Comment About Specific Learning Objects:

Any Other Comments::

II. Below is a list of the various learning objects that could have been assigned to you. Based on the short description, please select 5 learning objects that you feel would have been the best to assign to you (you may include learning objects you saw).

Learning Object #	Description
1	Text article. Overview of autism, early signs, causes, treatment
2	Text article. Facts about autism, funding of autism (very brief bullet facts”)
3	Text article. Overview of diagnosis, early signs
4	Text article. Taken from Center for Disease Control. Early signs, symptoms
5	Text article. Personal reflections on sending a college age autistic daughter to university, including school selection, course selection, interactions with administration, peer tutoring and socialization.
6	Text article. Advice for new peer tutors: things that may upset an autistic child and why
7	Text article. Screening for autism, details about in-depth screening, other issues (such as hearing lose or elevated blood lead levels) that may accompany autism, setting up an IEP and services from the school system
8	Video. Individuals talk about their experiences with Autism, a mother talks about her non-verbal young son, and a teenager talks about his experiences coping with Autism - tantrums, parenting demands, socialization
9	Video. Mother talking about her experience raising her son, how stressful it was when he’d throw tantrums before he could talk and about her conversations with her other son about curing autism
10	Text article. Advice for getting through holidays and social gatherings
11	Text article. Advice for parents with a newly diagnosed child, with topics such as advocating, finding support groups, educating themselves and participating in research
12	Text article. Overview of Asperger’s, its history, symptoms and connection to autism and pdd.
13	Text article. DSM-IV definition of Asperger’s, differentiation between it and autism. Quite technical.
14	Text article. Informal overview of Asperger’s syndrome, common behaviours and parenting strategies. Longer than most of the other objects (about 6.5 minutes)
15	Text article. Coming to terms with the diagnosis of an autism spectrum disorder for a child. Advice for caregivers to take care of themselves.
16	Text article. Tips for parents, siblings, and extended family (such as grandparents) of a child with autism
17	Video. Temple Grandin is introduced. She discusses autism, symptoms and behaviours and raising children on the spectrum.
18	Video. Temple Grandin where she discusses sensory issues, accomodations and punishing bad behaviour (but not sensory issues)
19	Text article. Advice on assembling and running a team (therapists, MDs, etc).
20	Text article. Unique abilities and other health concerns that may accompany autism.

III. After looking at the example annotated learning objects and having the concept of annotating explained to you.

1. Do you find any value in using learning objects with annotations?

-5	-4	-3	-2	-1	0	1	2	3	4	5
less value					neutral					more value

2. How likely would you be to contribute annotation to a learning object if using a system that supported this?

-5	-4	-3	-2	-1	0	1	2	3	4	5
unlikely					neutral					likely

3. How often might you leave annotations?

0	1	2	3	4	5	6	7	8	9	10
never										always

4. How satisfied would you be reading annotations left by previous students?

-5	-4	-3	-2	-1	0	1	2	3	4	5
unsatisfied					neutral					satisfied

Appendix D

Recruitment Letter

Department Letterhead
University of Waterloo
Date

I am looking for people willing to participate in a study that is investigating the design of effective intelligent tutoring in the domain of home healthcare, targeting care of children with Autism Spectrum Disorder. We are looking to recruit any adult (18 years or older) who is a caregiver for a child with autism.

The study will be conduct in the AI laboratory, Room 2306C in the Davis Centre at the University of Waterloo, or at a location that is convenient for you.

The study in its entirety will take approximately 60 minutes to complete and you will receive a $20 Chapter's Gift Certificate in appreciation of your participation.

Participation in the study involves a one hour session. If you choose to participate you will be given a short, multiple-choice assessment (pertaining to knowledge of autism spectrum disorders), then provided with a learning object (which can be completed in 5-10 minutes) appropriate to this assessment. This will be repeated 5 times, with an assessment at the completion of each of the five learning objects (pertaining to knowledge of autism spectrum disorders), a form to complete concerning streamlining the object and a final assessment after viewing all five learning objects. At the end of the session you will be asked to complete a questionnaire concerning satisfaction with the learning objects. All information you provide will be recorded anonymously.

This study may contribute to the advancement of the artificial intelligence subfield of the design of effective intelligent tutoring systems, home healthcare and care giving for children with Autism Spectrum Disorder.

If you are interested in participating, please respond to this e-mail and provide times that you would be available to participate within one of the time frames specified below. I will send you a letter outlining the study and a scheduled time for participation.

We are interested in having about 5 participants for this study. Should you have any questions about the study, please contact either John Champaign at jchampai@uwaterloo.ca or my faculty supervisor Robin Cohen at rcohen@uwaterloo.ca. Further, if you would like to receive a copy of the results of this study, please contact either investigator. Please note that this study has been reviewed by the Office of Ethics Research at the University of Waterloo and has received ethics clearance.

Date ***Time*** ***Place***
Date ***Time*** ***Place***
Date ***Time*** ***Place***
Date ***Time*** ***Place***
Date ***Time*** ***Place***

John Champaign
PhD Candidate
David R. Cheriton School of Computer Science
jchampai@uwaterloo.ca

Appendix E

Multi-Dimensional View of Human Study Assessment

For our human study (see Chapter 7) there was a challenge in applying the multi-dimensional model of knowledge (see Sections 2.6.4.1 and 3.2). Unlike the case of our simulations, where we could directly model and measure the simulated students' understanding of various knowledges, with human students we only had the answers to quiz questions to use for our calculation of student similarity. The domain of instruction, care of children on the autism spectrum, was inherently complex and determining what abilities make up the overall knowledge of the course of study and categorizing assessment questions according to this ontology would have been challenging.

Our simplifying assumption was that each question on the quiz represented a dimension in the model of knowledge for the students. This gave us a 10 dimensional model (one for each question), and the similarity of students could be matched based on how they answered each question, regardless of whether that answer was correct or not. For example two students who both incorrectly answered "speech therapy" to question 9 would be more similar than a student who answered physical therapy (which was also incorrect).

It was possible for us to do this since there was only a single quiz for assessment for the entire experiment. Had we used multiple quizzes, it would not have been possible to compare the results in this manner (they would be in different vector spaces). In this case there would have been a need to map the assessments to a global knowledge, which could then have been compared.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

[2] Hua Ai, Rohit Kumar, Dong Nguyen, Amrut Nagasunder, and Carolyn Rosé. Exploring the effectiveness of social capabilities and goal alignment in computer supported collaborative learning. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 134–143. Springer Berlin / Heidelberg, 2010.

[3] David Akers. Wizard of Oz for participatory design: inventing a gestural interface for 3D selection of neural pathway estimates. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06, pages 454–459, New York, NY, USA, 2006. ACM.

[4] R. B. Almeida, B. Mozafari, and Junghoo Cho. On the Evolution of Wikipedia. International Conference on Weblogs and Social Media, March 2007.

[5] Ryan S.J.d. Baker. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 1059–1068, New York, NY, USA, 2007. ACM.

[6] Scott Bateman, Rosta Farzan, Peter Brusilovsky, and Gord McCalla. Oats : The open annotation and tagging system. *Information Sciences*, 2006.

[7] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

[8] Helen Bretzke and Julita Vassileva. Motivating cooperation on peer to peer networks. In *User Modeling*, pages 218–227, 2003.

[9] Amber L. Briggs and Susan Cornell. Self-monitoring Blood Glucose (SMBG): Now and the Future. *Journal of Pharmacy Practice*, 17(1):29–38, 2004.

[10] Christopher Brooks and Gord McCalla. Towards flexible learning object metadata. In *International Journal Of Continuing Engineering And Lifelong Learning*, pages 50–63, 2006.

[11] Christopher A. Brooks, Rupi Panesar, and Jim E. Greer. Awareness and collaboration in the iHelp courses content management system. In *EC-TEL*, pages 34–44, 2006.

[12] Peter Brusilovsky and Nicola Henze. Open corpus adaptive educational hypermedia. In *The Adaptive Web*, pages 671–696, 2007.

[13] Peter Brusilovsky and J. Vassileva. Course sequencing for large-scale web-based education. *International Journal of Continuing Engineering Education and Life-long Learning*, 13(1/2):75–94, 2003.

[14] Paul Buitelaar and Bernardo Magnini. Ontology learning from text: An overview. In *In Paul Buitelaar, P., Cimiano, P., Magnini B. (Eds.), Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press, 2005.

[15] Luca Camerini, Michele Giacobazzi, Marco Boneschi, Peter J. Schulz, and Sara Rubinelli. Design and implementation of a web-based tailored gymnasium to enhance self-management of fibromyalgia. *User Modeling and User-Adapted Interaction*, 21(4-5):485–511, 2011.

[16] J.R. Carbonell. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *Man-Machine Systems, IEEE Transactions on*, 11(4):190 –202, dec. 1970.

[17] J.B. Carroll and S. Sapon. *Modern language aptitude test MLAT: manual.* Second Language Testing, 2002.

[18] Ran Cheng and Julita Vassileva. Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Modeling and User-Adapted Interaction*, 16(3-4):321–348, 2006.

[19] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):pp. 493–507, 1952.

[20] David N. Chin. User modeling in UC, the UNIX consultant. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '86, pages 24–28, New York, NY, USA, 1986. ACM.

[21] Nathalie Colineau and Cécile Paris. A portal to promote healthy living within families. In *eHealth*, pages 259–266, 2010.

[22] Nathalie Colineau and Cécile Paris. Motivating reflection about health within the family: the use of goal setting and tailored feedback. *User Modeling and User-Adapted Interaction*, 21(4-5):341–376, 2011.

[23] Michel C. Desmarais and Ildikó Pelczer. On the faithfulness of simulated student performance data. In *Educational Data Mining*, pages 21–30, 2010.

[24] Toby Dragon, Mark Floryan, Beverly Woolf, and Tom Murray. Recognizing dialogue content in student collaborative conversation. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 113–122. Springer Berlin / Heidelberg, 2010.

[25] Jonathan Foss and Alexandra Cristea. Transforming a linear module into an adaptive one: Tackling the challenge. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 82–91. Springer Berlin / Heidelberg, 2010.

[26] Malcolm Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference.* Abacus, February 2002.

[27] Joshua Gorner and Robin Cohen. Optimizing advisor network size in a personalized trust-modelling framework for multi-agent systems. In *Thirteenth International Workshop on Trust in Agent Societies (TRUST-2010)*, Toronto, 2010, 2010.

[28] Arthur Graesser, Patrick Chipman, Brandon King, Bethany McDaniel, and Sidney D'Mello. Emotions and learning with AutoTutor. In *Proceeding of the 2007 Conference on Artificial Intelligence in Education*, pages 569–571, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.

[29] Mark Granovetter. The strength of weak ties: A network theory revisited. *Sociological Theory*, 1(1983):201–233, 1983.

[30] Jim Greer and Gord McCalla. *Student Modelling: The Key to Individualized Knowledge-Based Instruction.* Springer, 1994.

[31] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human Computer Studies*, 43(5-6):907–928, 1995.

[32] Andreas Harrer, Bruce M. McLaren, Erin Walker, Lars Bollen, and Jonathan Sewall. Creating cognitive tutors for collaborative learning: steps toward realization. *User Modeling and User-Adapted Interaction*, 16:175–209, September 2006.

[33] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information and System Security (TISSEC)*, 22(1):5–53, 2004.

[34] Jeff Howe. The rise of crowdsourcing. *Wired*, 14(6), 2006.

[35] Johanna Höysniemi, Perttu Hämäläinen, and Laura Turkki. Wizard of Oz prototyping of computer vision based action games for children. In *Proceedings of the 2004 conference on interaction design and children: building a community*, IDC '04, pages 27–34, New York, NY, USA, 2004. ACM.

[36] Seiji Isotani, Riichiro Mizoguchi, Sadao Isotani, Olimpio Capeli, Naoko Isotani, and Antonio de Albuquerque. An authoring tool to support the design and use of theory-based collaborative learning activities. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 92–102. Springer Berlin / Heidelberg, 2010.

[37] G. Tanner Jackson and Arthur C. Graesser. Content matters: An investigation of feedback categories within an ITS. In *Proceeding of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 127–134, Amsterdam, The Netheralnds, The Netherlands, 2007. IOS Press.

[38] T. S. Jastrzembski and K. A. Gluck. A formal comparison of model variants for performance prediction. In *In Proceedings of the 9th International Conference of Cognitive Modeling*, Manchester, United Kingdom, 2009.

[39] W. Lewis Johnson and Elliot Soloway. PROUST: Knowledge-based program understanding. In *International Conference on Software Engineering*, pages 369–380, 1984.

[40] W. Lewis Johnson and Andre Valente. Tactical language and culture training systems: using artificial intelligence to teach foreign languages and cultures. In *Proceedings of the 20th national conference on Innovative applications of artificial intelligence - Volume 3*, IAAI'08, pages 1632–1639. AAAI Press, 2008.

[41] Imène Jraidi and Claude Frasson. Subliminally enhancing self-esteem: Impact on learner performance and affective state. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 11–20. Springer Berlin / Heidelberg, 2010.

[42] Judy Kay. Lifelong learner modeling for lifelong personalized pervasive learning. *IEEE Transactions on Learning Technologies (TLT)*, 1(4):215–228, 2008.

[43] J. F. Kelley. An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '83, pages 193–196, New York, NY, USA, 1983. ACM.

[44] J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information and System Security (TISSEC)*, 2:26–41, January 1984.

[45] Cynthia Kersey, Barbara Di Eugenio, Pamela Jordan, and Sandra Katz. KSC-PaL: A peer learning agent. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 72–81. Springer Berlin / Heidelberg, 2010.

[46] Nicolas Labeke, Alexandra Poulovassilis, and George Magoulas. Using similarity metrics for matching lifelong learners. In *ITS '08: Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, pages 142–151, Berlin, Heidelberg, 2008. Springer-Verlag.

[47] H. Lane, Mike Schneider, Stephen Michael, Justin Albrechtsen, and Christian Meissner. Virtual humans with secrets: Learning to detect verbal cues to deception. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 144–154. Springer Berlin / Heidelberg, 2010.

[48] John Lee, Finbar Dineen, and Jean McKendree. Supporting student discussions: it isn't just talk. *Education and Information Technologies*, 3:217–229, December 1998.

[49] Roberto S. Legaspi, Raymund Sison, and Masayuki Numao. A category-based self-improving planning module. In *Intelligent Tutoring Systems*, pages 554–563, 2004.

[50] Blair Lehman, Sidney D'Mello, and Natalie Person. The intricate dance between cognition and emotion during expert tutoring. In Vincent Aleven, Judy Kay, and

Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg, 2010.

[51] Cecil Alec Mace. *Psychology of study.* Methuen & Co. Ltd., 1932.

[52] Catherine C. Marshall. Toward an ecology of hypertext annotation. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, HYPERTEXT '98, pages 40–49, New York, NY, USA, 1998. ACM.

[53] Noboru Matsuda, William W. Cohen, Jonathan Sewall, Gustavo Lacerda, and Kenneth R. Koedinger. Evaluating a simulated student using real students data for training and testing. In *User Modeling*, pages 107–116, 2007.

[54] Gord McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education*, 7:1–23, 2004.

[55] Christopher A. Miller, Peggy Wu, and Harry B. Funk. A computational approach to etiquette: Operationalizing Brown and Levinson's politeness model. *IEEE Intelligent Systems*, 23(4):28–35, 2008.

[56] Patrice Moguel, Pierre Tchounikine, and André Tricot. Supporting learners self-organization: An exploratory study. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 123–133. Springer Berlin / Heidelberg, 2010.

[57] R. Morales and A.S. Aguera. Dynamic sequencing of learning objects. pages 502–506, 2002.

[58] Tom Murray. Authoring intelligent computer systems: An analysis of the state of the art. *International Journal of Artifical Intelligence in Education*, 10:98–129, 1999.

[59] Stellan Ohlsson. The interaction between knowledge and practice in the acquisition of cognitive skills. In A. Meyrowitz and S. Chipman, editors, *Foundations of knowledge acquisition: Cognitive models of complex learning*, pages 147–208. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1993.

[60] Cécile Paris and Stephen Wan. Capturing the user's reading context for tailoring summaries. In *UMAP*, pages 337–342, 2009.

[61] B. Peintner, P. Viappiani, and N. Yorke-Smith. Preferences in interactive systems: Technical challenges and case studies. *AI Magazine*, 29(4):13–24, Winter 2008.

[62] Paul Pimsleur. A memory schedule. *The Modern Language Journal*, 51(2):pp. 73–75, 1967.

[63] D. Plant. hSITE: healthcare support through information technology enhancements. *NSERC Strategic Research Network Proposal*, 2008.

[64] Mian Qin, Scott Buffett, and Michael W. Fleming. Predicting user preferences via similarity-based clustering. In *Canadian Conference on AI*, pages 222–233, 2008.

[65] Neil Rambo and Christine Beahler. Knowledge-based information and systems. In Patrick W. OCarroll, William A. Yasnoff, M. Elizabeth Ward, Laura H. Ripp, and Ernest L. Martin, editors, *Public Health Informatics and Information Systems*, Health Informatics, pages 352–375. Springer New York, 2003.

[66] Timothy Read, Beatriz Barros, Elena Bárcena, and Jesús Pancorbo. Coalescing individual and collaborative learning to model user linguistic competences. *User Modeling and User-Adapted Interaction*, 16(3-4):349–376, 2006.

[67] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[68] Robert A. Richards. Principle hierarchy based intelligent tutoring system for common cockpit helicopter training. intelligent tutoring systems. In *Proceedings of ITS 2002*, pages 473–483, 2002.

[69] Linda Rosa, Emily Rosa, Larry Sarner, and Stephen Barrett. A close look at therapeutic touch. *JAMA: The Journal of the American Medical Association*, 279(13):1005–1010, April 1998.

[70] Andre A. Rupp, Shauna J. Sweet, and Younyoung Choi. Modeling learning trajectories with epistemic network analysis: A simulation-based investigation of a novel analytic method for epistemic games. In *Educational Data Mining*, pages 319–320, 2010.

[71] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*, chapter Informed Search and Exploration, pages 106–109. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

[72] Aaditeshwar Seth, Jie Zhang, and Robin Cohen. A multi-disciplinary approach for recommending weblog messages. In *National Conference on Artificial Intelligence (AAAI08) Workshop on Enhanced Messaging*, 2008.

[73] Aaditeshwar Seth, Jie Zhang, and Robin Cohen. A subjective credibility model for participatory media. In *National Conference on Artificial Intelligence (AAAI08) Workshop on on Intelligent Techniques for Web Personalization & Recommender Systems*, 2008.

[74] Aaditeshwar Seth, Jie Zhang, and Robin Cohen. Bayesian credibility modeling for personalized recommendation in participatory media. In Paul De Bra, Alfred Kobsa, and David Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin / Heidelberg, 2010.

[75] Joseph B. South and David W. Monson. A university-wide system for creating, capturing, and delivering learning objects. In *The Instructional Use of Learning Objects: Online Version*, 2000.

[76] H.F. Spitzer. Studies in retention. *Journal of Educational Psychology*, 30(9):641 – 656, 1939.

[77] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, March 2006.

[78] United Nations Educational, Scientific and Cultural Organization Institute for Statistics. Global Education Digest, 2007.

[79] Kurt VanLehn. *Mind Bugs – The Origins of Procedural Misconceptions.* MIT Press, 1990.

[80] Kurt VanLehn, Stellan Ohlsson, and Rod Nason. Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education*, 5:135–175, 1996.

[81] Julita Vassileva. Dynamic course generation. *Journal of Computing and Information Technology*, 5:87–102, 1997.

[82] Julita Vassileva. DCG + GTE: Dynamic courseware generation with teaching expertise. *Instructional Science*, 26 (3/4):317–332, 1998.

[83] Julita Vassileva. Goal-based autonomous social agents: Supporting adaptation and teaching in a distributed environment. In *Intelligent Tutoring Systems*, pages 564–573, 1998.

[84] Julita Vassileva. Toward social learning environments. *IEEE Transactions on Learning Technologies (TLT)*, 1(4):199–214, 2008.

[85] Julita Vassileva and Ralph Deters. Dynamic courseware generation on the WWW. *British Journal of Educational Technologies*, 29 (1):5–14, 1998.

[86] Julita Vassileva, Gordon McCalla, and Jim Greer. Multi-agent multi-user modeling in I-Help. *User Modeling and User-Adapted Interaction*, 13:179–210, February 2003.

[87] Chieu Vu Minh, Vanda Luengo, and Lucile Vadcard. A framework for building intelligent learning environments in ill-defined domains. In *13th International Conference on Artificial Intelligence in Education. Workshop AIED Applications in Ill-Defined Domains*, page 5, 2007.

[88] Xin Wan, Q. Jamaliding, and T. Okamoto. Discovering social network to improve recommender system for group learning support. In *International Conference on Computational Intelligence and Software Engineering (CiSE)*, pages 1 –4, 2009.

[89] Claire Williams and Sidney D'Mello. Predicting student knowledge level from domain-independent function and content words. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 62–71. Springer Berlin / Heidelberg, 2010.

[90] Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 73–80, 2003.

[91] Jie Zhang and Robin Cohen. A personalized approach to address unfair ratings in multiagent reputation systems. In *AAMAS'06 Workshop on Trust in Agent Societies*, 2006.

[92] Jie Zhang and Robin Cohen. A comprehensive approach for sharing semantic web trust ratings. *Computational Intelligence*, 23(3):302–319, 2007.

[93] Jie Zhang and Robin Cohen. Design of a mechanism for promoting honesty in e-marketplaces. In *AAAI Conference on Artificial Intelligence*, pages 1495–1500, 2007.

[94] Jie Zhang and Robin Cohen. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, 7(3):330–340, 2008.

[95] Amal Zouaq, Roger Nkambou, and Claude Frasson. Building domain ontologies from text for educational purposes. In *European Conference on Technology Enhanced Learning (EC-TEL)*, pages 393–407, 2007.