UNIVERSITY OF CALIFORNIA,
IRVINE

Supporting Large-Scale Engineering Education:
the Active Learning Personal Advisor via Course Automation

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by

Quoc-Viet Pham Dang

Dissertation Committee:
Professor Daniel Gajski, Chair
Professor Rainer Doemer
Professor Fadi Kurdahi

2017

DEDICATION

To

my family and friends

for their patience and support

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First off, I would like to extend my deepest gratitude to my committee members, Professor Daniel Gajski, Professor Rainer Doemer, and Professor Fadi Kurdahi, who have all supported me and shared their knowledge and insight into both industry and academia over the years. I have been very fortunate to be able to be a “fly on the wall” during their many lunch conversations for more than the past decade. I look forward for an opportunity to continue to work with each of them moving forward. Without them, none of this would be possible.

I would also like to acknowledge my family, consisting of my siblings, parents, and grandparents. They have continually encouraged me to pursue a higher education and have been very patient with me and supporting me every step of the way. I would like to specifically thank my mother for providing me several thousand packed meals during my time as a graduate student so I could focus on studying instead of preparing meals.

Most of all, I would like to thank again, my committee chair, Professor Daniel Gajski. I first met Dr. Gajski when I interviewed for a work study position at UCI about 15 years ago, which I almost didn’t get. I didn’t realize it back then, but meeting Dr. Gajski was one of the defining moments of my life. He has patiently advised, guided, and supported me for almost my entire adult life. I have no idea what I would be doing today had our paths not crossed. I hope to one day live up to the potential he has always seen in me.

Last but definitely not least, financial support for this research was provided in part by University of California, Irvine and NSF Grant #1136146.

CURRICULUM VITAE

Quoc-Viet Pham Dang

2004	B.S. in Information and Computer Science, University of California, Irvine
2004-10	Programmer Analyst, University of California, Irvine
2010	M.S. in Computer Engineering and Computer Science, California State University, Long Beach
2011-13	Teaching Assistant / Graduate Student Researcher University of California, Irvine
2013-17	Teaching Associate / Graduate Student Researcher University of California, Irvine
2017	Ph.D. in Computer Engineering, University of California, Irvine

FIELD OF STUDY

Computer Systems and Software

PUBLICATIONS

Daniel Gajski, Quoc-Viet Dang, and WenLiang He, "An Online Methodology for Individualized Education," Paper presented at International Conference on e-Learning, e-Business, EIS, & e-Government (EEE'13).

Quoc-Viet Dang and Daniel Gajski, "Bringing In-Class Online: A Hybrid Solution," Paper presented at 4th Interdisciplinary Engineering Education Conference (IEDEC 2014).

Quoc-Viet Dang and Daniel Gajski, "Creating an Automated Learning Management Tool in an Engineering Course Based on Interdisciplinary Metrics," Paper presented at 2014 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS 2014).

ABSTRACT OF THE DISSERTATION

Supporting Large-Scale Engineering Education through Development of an Automated Learning Management Tool: the Active Learning Personal Advisor.

By

Quoc-Viet Pham Dang

Doctor of Philosophy in Computer Engineering

University of California, Irvine, 2017

Professor Daniel Gajski, Chair

Public education is on the brink of a potential crisis attempting to significantly increase student enrollment while maintaining quality of education. Online courses have been proposed and debated among members of the UC regents, numerous college administrators, faculty, and students. On one hand, online education can reduce overhead while enrolling more students. Directly translating the classroom lectures and materials to an online environment does not necessarily produce equivalent student performance and satisfaction from the course compared to an in-class environment. Since there is no universal standard for online education, erratic and inconsistent results have been achieved in terms of student performance and costs to students as well as administration. A hybrid scalable teaching and learning methodology is required by both educators and students to achieve the greatest advantages of using today's technology and to apply it toward improving student performance and participation.

This dissertation presents a methodology and system to provide a more individualized and responsive learning environment for students in large hybrid and online university courses while keeping overall costs and time commitment down as well as improve overall student

performance. The Active Learning Personal Advisor, the implemented learning design tool of this research, is developed based on multi-disciplinary metrics and studies from the fields of Psychology, Education, and Engineering. A primary limiting resource for both students and instructors is time. By automating some basic key interactions that may occur between students and instructors, hours of each individual's time can be saved, maximizing the quality of the available in-person interactions to occur during a course while allowing for a more scalable sized classroom environment.

INTRODUCTION

Public education is on the brink of a crisis. The University of California has increased admissions from 50, 291 in Fall 2003 to 82,850 for Fall 2013 [1] [2]. Student-faculty ratios and number of credit hours per faculty member have increased 10% in the last several years alone amidst UC budget cuts [3]. These increases present a problem for running large classes. Typical classroom environments do not scale well as the student to instructor ratio increases. Furthermore, directly translating the classroom lectures and materials to an online environment in order to support even more students does not necessarily produce equivalent student performance and satisfaction from the course compared to a smaller in-class environment.

Since there is no standard set in place for online education, student performance and costs associated with various offerings are inconsistent. Many Massive Open Online Courses (MOOCs) are still teaching in a scaled up version of the typical in-class methodology. The primary focus of many online course sites is to provide content, such as courses offered by Coursera [4], Udacity [5], and other MOOCs. Lecture and course content, along with plenty of practice material is available but only fully utilized by self-motivated students. According to a national survey conducted across 560 colleges and universities, approximately 1 out of 5 incoming and first year undergraduate students do not know how to properly study or prepare for college level courses by themselves [6]. Providing more content with no guidance does not maximally aid students. This methodology of producing more and more content in its present state does not work well for larger courses. Translating this method into an online environment makes the situation even worse since there is even less in-person interaction allocated per student. For example, San Jose State University offered a course for approximately 100,000 students in collaboration with Udacity in 2013 [7]. Unfortunately, many students did not receive the help they needed, and many students failed or did not complete the course.

In a small preliminary experiment conducted in conjunction with the School of Education at UCI in [8] shows that although the top 1/3 of students tends to still do well with an in-class methodology taught in an online environment, the other 2/3's of students do not perform as well when exposed to the same situation; in fact, the bottom 1/3 of students did much worse relative to the top 1/3 of students. Different students respond to and perform differently when exposed to different teaching and learning styles [9] [10] [11]. Using information from this research and combining it with other research theories and methodologies, a specialized design learning tool can be developed to facilitate and automate the changes required to allow instructors to manage massive online courses significantly better than current available options [12].

A hybrid scalable teaching and learning methodology is required by both educators and students to achieve the greatest advantages of using today's technology and to apply it toward improving student performance and participation [13]. Any process or portion of this methodology that can be automated will save meaningful amounts of time for both students and instructors, allowing for higher level of critical thinking and applications to occur with the time now made available.

Chapter 1: The Inevitability of Larger Classrooms

The traditional in-class methodology was developed for small classrooms of 15-20 students. Low student to teacher ratios, typically under 20 students per teacher, have been preferred and recommended to maximize student achievement, engagement, and retention from research starting in the 1970's [14] [15] [16]. Actual classroom sizes for K-12 vary depending on a variety of factors [17]. Today, some undergraduate Engineering courses consist of more than ten times that many students: some who are interested, some who just want a passing grade, and others who are not yet ready for college and do not properly prepare to study material. In fact, according to a national survey consisting of 560 colleges and universities in 2016, 20% of first-year college students had difficulty learning and getting help with coursework [6] [18]. As classroom sizes increase and varying levels of experiences of students come into play, this situation will only exacerbate existing problems and deficiencies utilizing current teaching methodologies and tools. The amount of time instructors and TAs have allocated to grade student work and guide students studies is limited and typically fixed per course offering.

On top of all of this, the UC Regents and college administration are moving toward massive online courses in order to increase income and minimize overhead costs in facilities. The UC President, Janet Napolitano is stressing for boosting undergraduate college student enrollment across all UC's by 10,000 by 2018 [19]. In order to increase income, the number of instructors hired by this time is not equivalent to maintain the instructor-to-student ratio, necessitating some kind of intervention method if quality of education is to remain the same or improve. The typical implementations of many of these online courses effectively make the courses the equivalent of online textbooks available at a library. Students are expected to study and understand explicit and tacit material by independently studying online content, working on homework, and finally take tests.

Recently, San Jose State University suspended its online education project with Udacity due to more than half of the students enrolled failing their final exams [7]. A possible erroneous assumption here by course developers was that students would be self-motivated and self-driven to do further research on their own when concepts were not clear. Perhaps an even bigger assumption was that developers assumed that most or all students would know what to search for when concepts didn't make sense; this is clearly not the case for a large number of students, as shown by the National Survey of Student Engagement for 2016 [6]. Even though online teaching and distance learning have been around for years, colleges and large companies are still having problems today with the most important metric: maintaining student performance with increased enrollment. In order to achieve this goal, current methodologies must be revised, and new tools or systems be developed to help support both educators and students.

A large percentage of difficulty of maintaining student performance can stem mainly from 2 sources:

1. Replicating a typical in-class methodology and applying it to an online or hybrid environment without making the necessary changes to ensure students stay on track throughout the course.
2. Not enforcing or verifying that students are actually following the outlined methodology for successfully completing the course.

The typical in-class methodology in its present state does not work well for larger courses. Translating this method into an online environment makes the situation even worse. Preliminary analysis [8] shows that the top 1/3 of students tends to still do well with an in-class methodology taught in an online environment, while the other 2/3's of students do not perform as well when exposed to the same situation; in fact, the bottom 1/3 of students did noticeably worse relative to the top 1/3 of students.

The group as a whole can benefit from changes in the teaching methodology. Students are also not following the ideal methodology, based on the survey results gathered from the preliminary analysis. With that in mind, some additional checks and verifications are needed in a methodology to ensure students stay on track. The design tool presented in this dissertation aims to address the following perspectives:

1. Student perspective: how to get the most out of their classroom experience with minimal investment in time.
2. Instructor/Institutional perspective: how to give the most quality of education per student using the limited (and shrinking) amount of time available.

In order to analyze these factors, current conventional methodologies are reviewed in order to develop a scalable method that addresses the above concerns. The design tool is developed to aid in implementation of the new methodology, saving time for both students and instructors alike. As motivational perspective to show the potential overall savings of automating minor interactions, take the case of a typical interaction between a student and instructor.

When a student gets stuck on a homework assignment or concept in the book or video, they may not look for the answer or know where to look [6]. In a typical case, the student may send an email to the TA or instructor for guidance. Usually the answers and recommendations are quite simple, but the wait time for a response can be anywhere in the span of hours up to a day, as observed in the Digital Design courses covered in this dissertation. Alternatively, if the student decides to wait until office hours, it still takes several minutes of interaction, pulling up the homework, book chapter, or online video slides in order to assess the situation. Assume that this interaction takes approximately 5 minutes to recommend a chapter section, example, or follow-up video/slide for the student to review. If 100 students had similar questions, 500 minutes of office hours or email support would have been spent answering just these simple questions. Even at just one question a week for each of these students, 5000 minutes would be spent in a 10 week course just giving recommendations for homework and example help. For larger sized classes, the situation gets exacerbated and eventually unmanageable.

An automated tool that can provide similar recommendations would free up all that time and allow for more meaningful discussions. Also, students would save hours individually in terms of getting stuck, waiting for responses, and then spending time to get back to where they were later when they got stuck. This is potentially even more beneficial for students who do not typically ask questions when they get stuck, hoping that attending lecture or discussion will answer their questions. The automated tool can pre-emptively suggest additional reading and viewing material, allowing the student to continue their studies without a long wait interval.

These lower level interactions are usually simple but are still needed because students may be stuck until their questions are answered much of the time. Reducing the turn-around time for these types of questions and automating the process can potentially save many hours of time for both students and instructors. These types of interactions, as well as other important situations where automated intervention is possible, are reviewed by considering current methodologies and practices. From there, a new hybrid methodology can be developed that more optimally utilizes a tool designed around automating some existing interactions.

Revising homework questions will take some initial overhead and implementation time. However, once homework questions are revised in a way that supports automated grading, that time spent by educators grading is now available to help students with more difficult concepts. In additional to automatically grading homework, pre-emptive recommendations can be generated as part of student feedback and comments versus waiting for students to ask questions regarding questions they missed, which students have reported to be helpful in the Digital Design courses offered in the past, as mentioned in part of the surveys available in the Appendix.

In-person office hours can also be improved through automated student assessment summaries, which highlight student strengths, weaknesses, and recommendations for further studying. With the amount of information available through homework, self-check quizzes, and individual exam questions, an automated system can provide insight into individual student performance that can be presented to the instructor and TAs when meeting with a student. These features can aid in saving time by narrowing down the areas discussed to figure out with what the student needs help.

Chapter 2: Current Conventional Methodologies

In general, current conventional methods focus on intervention and education models for small groups of students or on disseminating content efficiently to a large group of students. There is a disconnect between distribution and effectiveness, leading to scalability issues when attempting to utilize techniques that work in a small classroom environment for a much larger audience. On one hand, providing individualized support for students in a small class is very effective but is impractical for a larger class, even with just 100 students simply due to time constraints. On the other hand, providing all the content, lecture materials, and references students may need during a course for all the students is not ideal either since students may not know how to best cover all the material in a fixed amount of time, and therefore must wait for potentially long periods of time before getting in-class or online help from instructors.

Present In-Class Methodology

The present in-class methodology was developed many years ago when classes consisted of 15-20 students. The instructor was able to keep in touch with students and address individual weaknesses and strengths through tailoring discussions and answers to fit each student. Incoming college students are more diverse today and have different motivations for taking courses, ranging from actually being interested to just wanting a passing grade for a required course [8].

In this section, we review a typical in-class methodology (Figure 1) and how students actually apply it in reality in more detail. From [8], we see a typical methodology that involves:

1. 1-2 hours of lecture with general questions covering basic concepts,
2. 1-2 hours of homework
3. 1 hour of discussion where homework and more detailed questions covering some lecture concepts are asked

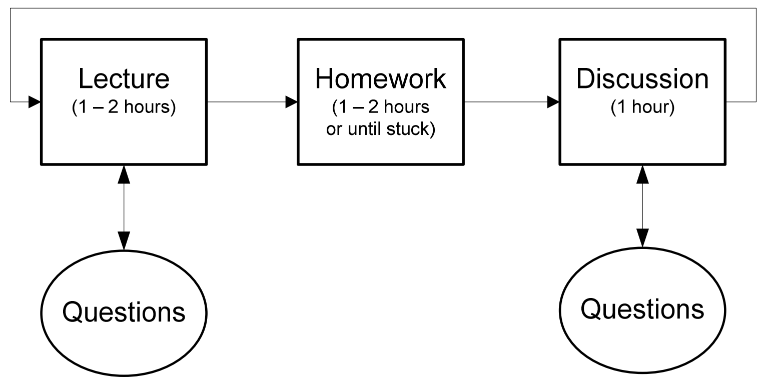

Figure 1: A Typical In-Class Methodology Work Flow.

These 3 basic steps are repeated each class session. This has a slow turn-around time for students who get stuck on some prerequisites and new concepts. In and of itself, this slow turn-around time is not detrimental in a small classroom. However, with 150+ students, the instructor cannot make the best use of lecture and discussion periods. Students who fall behind cannot catch up. With a massive online class, this methodology would create a dire situation for typical students.

To exacerbate the slow learning situation, survey results gathered from [8] indicate that students in the Digital Logic Engineering Course were not spending the expected amount of time outside of class studying and also don’t start on homework as early as they should have.

The majority of students from the Digital Logic course were following an even slower methodology (Figure 2) involving just attending either lecture or discussion and waiting until the last minute to start and complete their homework. This study pattern was observed and confirmed by student behavior during lecture and discussion sections.

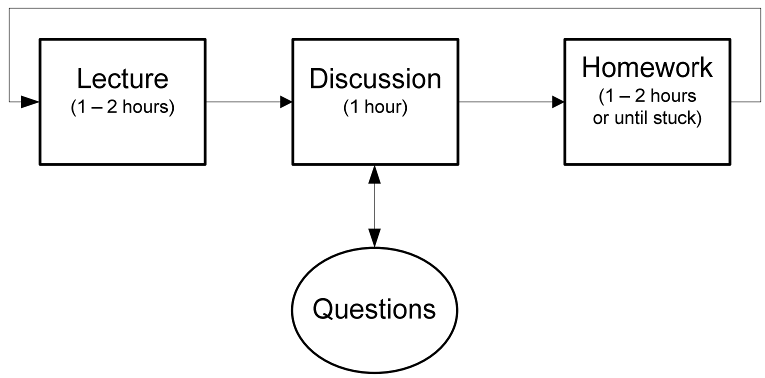

Figure 2: Student Implementation of In-Class Methodology Work Flow.

There were very few, if any, questions asked during lecture. Most of the simpler conceptual questions were asked during discussion. Students didn't start and complete homework until close to the due date. In many cases, students are unsure on what questions to ask, what materials to study, and in what order. Questions regarding harder concepts that were expected to be asked during discussion were now asked after homework was turned in at the next lecture or discussion section, if at all.

Usually, these harder conceptual questions were asked 1-2 or more lectures and discussions late. This was also confirmed by noticing that the timestamps for many electronic homework submissions, which were turned in on the last possible day even though the homework was available for at least a week in advance. This student methodology of learning perpetually made a majority of the students fall further and further behind in class as course progresses.

A fundamental change to the typical in-class methodology is required since classes are getting bigger and students' prior knowledge is more diversified and usually not adequate. Additional checks and verifications also have to be added to ensure that students stay on track during the course.

Present Online Based Methodology

The typical present online methodology expands upon the in-class methodology. Since the in-class methodology doesn't scale well, this is exacerbated with online courses which normally tend to have much higher enrollment. There is also a lack of individualized support for many online courses offered by colleges.

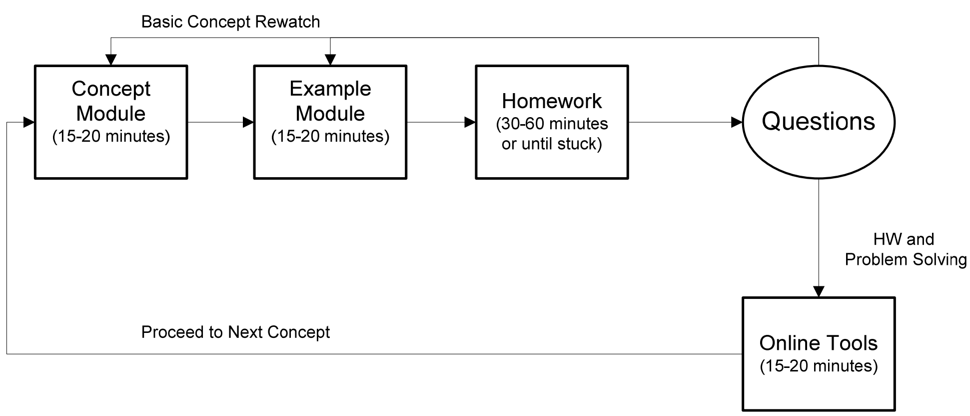

Figure 3: A Typical Online Methodology Work Flow.

An online methodology (Figure 3) is advantageous in minimizing the course overhead and can help only highly self-motivated students. Very few individuals can learn by themselves by reading textbooks or watching videos and taking exams. The basic on-line methodology lacks the individual support needed for the majority of students to be successful, as will be discussed in the Results section. For students who are not motivated or do not have adequate background knowledge, the use of concept modules, example modules, and online tools is less frequent, which reduces the effectiveness of these modules. A more individualized approach is needed.

Concept and Example Modules

Lecture and discussion sections typically last 1-2 or more hours and cover several concepts each session. As shown in [20], it can be difficult for students to maintain attention for more than 10-15 minute time spans at once, and in some cases even less; it is also dependent upon the teaching style of the instructor to maintain that attention several times during the lecture or discussion. By breaking down lectures into individual concepts, it is easier to keep students engaged and the overhead of keeping their attention is reduced since key recapturing techniques such as demonstrating concepts on a more concrete level, asking questions, and summarizing the concepts [20] are built into each 15-20 minute module. A flowchart of a concept module is shown in Figure 6 in the Front-End Interface Overview section.

The added convenience of being able to refer to a specific example or concept is also preferred by students [8]. The ability to more rapidly use and refer to these individual modules aids students in completing homework and having questions answered in a quicker fashion motivates students more than the typical in-class methodology they usually follow.

Depending on the current course layout, the overhead time involved converting 1-2 hour lecture modules into finalized concept modules can take up to 5-10 hours per lecture, as was the case with the Digital Logic Engineering Course we examined in [8]. The slides, the flow, the examples have to be redone.

The benefit is that the concept modules do not have to be updated all the time and can usually be kept between different offerings of the course. Whether there are 150 students, like the groups compared in the Digital Logic Engineering Course, or thousands of students, there is no additional overhead for normal lecture sessions in terms of requiring larger lecture halls. Being able to address individual student concerns are covered in the following sections.

Online Tools

If the concept and example modules are not enough to answer student questions, additional help is available through online tools such as message boards and email lists. Message boards and email lists have the potential to accrue minimal overhead and time commitment if students actively participate and help each other. Instructor and teaching assistant presence is still required; and their time involvement depends on the course structure, format, and difficulty of homework problems. A flowchart of the Online Tools usage from a student's perspective is shown in Figure 7 in the Front-End Interface Overview section.

Previous Works

The traditional in-class methodology was developed for small classrooms of 15-20 students [15]. Today, some undergraduate Engineering courses consist of more than ten times as many students: some who are interested, some who just want a passing grade, and others who are not ready for college and do not know how to properly prepare or study material. Extending the present typical methodology to such diverse experience levels does not work.

The typical present online methodology expands upon the in-class methodology. Since the in-class methodology doesn't scale well, this is exacerbated with online courses which normally tend to have much higher enrollment. There is also a lack of individualized support for many online courses offered by colleges.

Some current examples of online tools used for education include Udacity [5] and Coursera [4]. Both are very popular but also have their limitations. Udacity only offers certain courses and is completely self-driven with no start and finish dates. In cases where Udacity partners with a school, such as SJSU [7], the results were less than optimal with more than half the students failing the final exam. Udacity's proposal to support more students with more success was primarily to add additional teaching assistants. Although this has an opportunity to work, it is costly and not practical for university consideration in the long term due to already stretched budgets. Instead, restructuring the existing methodology has more potential to create scalable and affordable instruction while maximizing success.

Coursera has a structured course flow, but it regularly follows the traditional in-class methodology of learning, which doesn't work for most types of students. It also does not have any recommendation or customized follow up features, leaving many of the courses self-guided and not providing additional help or only minimal universal help when students get stuck.

MOOC's in general do not customize or adapt readily for the varying types and skills of students. Overall, the primary focus has been providing content, essentially creating a massive online library. Interactive tools and self-check quizzes exist, but there is no follow-up, much less personalized follow-up.

As we have seen from previous research, the success and satisfaction from taking such courses are completely based on student self-motivation and their own perception of themselves academically [21]. So, if a student does not have high confidence in their understanding of the current concept and does not receive any guidance, they will not likely be self-motivated to continue. If recommendation and support were more readily available, such as an automated recommendation at the end of a self-check quiz that dynamically adjusts its recommendations based on the current student's performance, the student may be more inclined to follow up with the resources presented, figure out what they did wrong, and in turn increase their own self-motivation and self-perception of their abilities.

Motivation for Improving Current Methods

Offering online courses can address the increasing number of students despite the shortage of large lecture halls, while generating some financial savings. However, faculty and students fear they will miss the face-to-face student-lecturer interaction in and out of the classroom. This is typically the case when schools increase enrollment since faculty-to-student ratio is not maintained. Of course, with some traditional in-class courses, having hundreds of students with varying backgrounds and motivation, the student-lecturer interaction has already been severely diminished from smaller classroom sizes, which are no longer feasible due to the shrinking educational budget. Since there is no standard in online education, it has produced erratic results in terms of student performance and costs to students as well as administration.

However, online tools, if managed and prepared for properly, have the opportunity to provide the highly regarded individualized learning experience of the small classroom with the lower costs of large lecture halls while decreasing overall overhead costs. In fact, the tools used for online courses can be utilized to improve the existing traditional classroom dynamic [22]

[23]. Students are trained from high school to prepare for testing by memorizing as many facts and examples as possible in order to answer each question. On the other hand, college professors tend to follow a top-down approach in teaching where they cover concepts, some math or algorithms, and methods or processes of finding proper solutions.

Technology has made vast amounts of information available for students. Unfortunately, it may have also inadvertently made too much information available; this makes it hard for students to know which resources should take precedent or sometimes even for what to search [6]. When used effectively, having access to large amounts of information through technology helps create a more engaged campus with higher retention and graduation rates [24]. However, as noted in Chapter 1, one out of five incoming college students do not know how to properly prepare or study for their courses [6] [18].

These processes, especially in Engineering, must be understood and not just memorized because they have many steps involved: specification, analysis, design and synthesis, optimization, verification, implementation, and testing. In a small college classroom of 20-30 students, lecturers can help students who fall behind individually by expending their out-of-class time. In a larger class of 150 students or more, the small classroom teaching style doesn't work as well since the time commitment for both the lecturer and students are dramatically increased for the same duration of time. The next generation of online tools to further education should focus on facilitating an individualized experience allowing the best use of available time for users and instructors.

The tools and methodology presented in this dissertation aid in providing the small classroom experience in a much larger setting while keeping overall costs and time commitment down. Such tools and methodology can be applied to any learning environment, not just online, to better prepare students for learning concepts and solving problems. It must cover (a) lectures, (b) sample problems with solutions, (c) questions and answers, (d) homework, and (e) tests. This is covered in more detail in [8].

The tools built to provide automated support follow a hybrid methodology, reviewed in the following chapter, which focuses on utilizing the successful aspects of both in-class and online-based methodologies in hopes of facilitating a “small-class feel” in a large classroom environment. The primary automated recommendations are through recommendations from self-check quiz follow-up recommendations and homework feedback comments. Students in the Digital Design course have found comments helpful, even when graders are limited on time and typically only able to provide generic comments. An automated system can recommend more specific chapters, videos, or other available resources that may relate to the student’s work. These automated recommendations save time for both students and educators.

In addition to help students save time, the automated system can generate reports per student regarding their performance in the course to allow an instructor or TA to quickly review a student’s strengths and weaknesses. These reports will allow an educator to quickly assess a student’s needs and help the educate guide the student appropriately without requiring a lengthy discussion beforehand that is typically required to assess a student’s current performance.

Chapter 3: Active Learning Personal Advisor Methodology Overview

Different students respond to and perform differently when exposed to different teaching and learning styles. Using information from this research and combining it with other research theories and methodologies mentioned in previous chapters, a specialized active learning tool has been developed to facilitate and automate the changes required to allow instructors to manage massive online courses significantly better than current available options.

The primary advantage of small in-class methodologies is the ability to offer personalized advice for each student. The main advantages of large online-based methodologies are throughput and content availability. However, in-class methods do not scale well to large classroom environments since the time allotment for in-person interaction does not increase linearly with classroom sizes. For online-based methods, individualized help is difficult because of large instructor to student ratios; furthermore, although there is usually plenty of content available, it can be overwhelming for students since there is little to no individualized guidance. A hybrid methodology combining the best practices of both methods can potentially address the shortcomings of both methods.

Hybrid Methodology

Small in-class based education does not work for large classes because of the diversity of students' knowledge, skills and motivation. Online learning alone does not work either since students are left alone to learn by themselves, which is similar to going to library, reading textbooks, and taking tests; this does not work for the majority of students. There is no teacher-student interaction except online message boards.

This design tool is based on a hybrid methodology (Figure 4) for the university environment that allows increased enrollment of several hundred students per course with only a small relative increase in overhead cost that also shows improved student performance levels over in-class and online. We are adding modules that are available to students who need extra help in the form of additional problem solving exercises, online tools, and discussion and tutoring groups. Lastly, homework feedback and follow-up emails are sent to students on a periodic basis through topic reviews to help reinforce what they have been learning.

Although online learning may provide some results, students currently still prefer an in-class learning environment [22] [9]; this is due mainly to online courses simply transferring existing lectures and discussions to an online environment and creating a poor perceived learning experience for students.

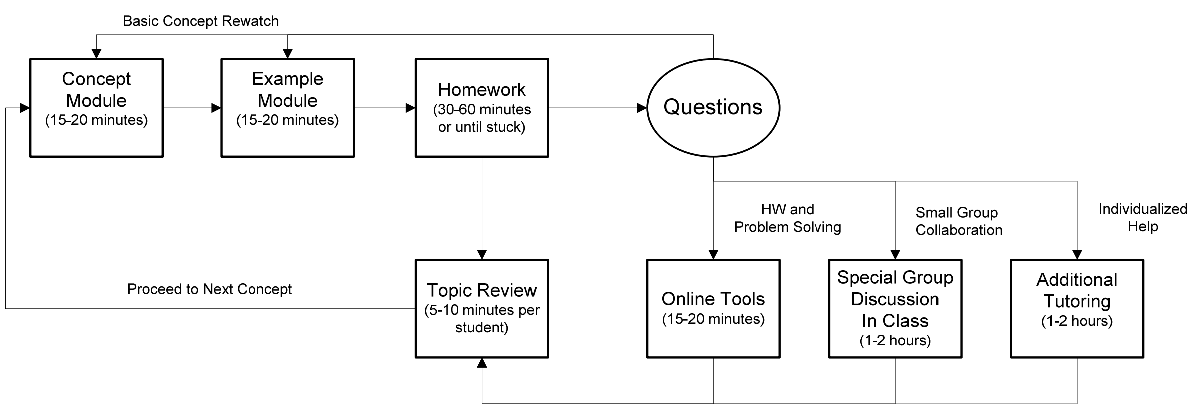

Figure 4: The Hybrid Methodology Work Flow.

There are many possible methodologies for both teaching and learning. In this section, we outline our proposed methodology to create an enhanced university style learning experience through hybrid instruction. Lectures and discussions are broken down into concept and example modules, respectively. Homework remains the same in this revision of the methodology.

The overall process flips the classroom [10] and has students use online concept and example modules to learn basic concepts while lectures and discussions are reserved for providing problem solving practice & clarification of those concepts while focusing on problem solving process.

In the hybrid model, lectures and discussions are replaced by Concept modules and Example modules. There are also modules that are available to students who need extra help in the form of additional problem solving exercises, online tools, and discussion and tutoring groups. Homework feedback and follow-up emails are sent to students on a periodic basis through topic reviews to help reinforce what they have been learning.

Having multiple guided materials for students to use allows multiple ways to assess a student's performance level, similar to strategies suggested in [25], can be used by the design tool to further guide students to the appropriate relevant materials when available. These modules will follow intelligent algorithms to make suitable teacher-student interaction possible even in very large scale classes and potentially automate much of the interaction.

Concept and Example Modules

Lecture and discussion sections typically last 1-2 or more hours and cover several concepts each session. As shown in [20], it can be difficult for students to maintain attention for more than 10-15 minute time spans at once, and in some cases even less; it is also dependent upon the teaching style of the instructor to maintain that attention several times during the lecture or discussion.

Concept and example modules can be more beneficial than typical lecture and discussion sections since they cover one concept at a time, can be viewed multiple times, and can be viewed at any time.

More importantly, self-check quiz questions may be available along with each concept and example module, which will allow students to actively learn and check their own understanding of the material. This is typically preferred over and more effective than simply re-reading notes or a chapter section, or re-watching a video module, just to study the material again without first checking their comprehension level of the material [25].

These modules can also stay the same between different offerings of the course, providing the added benefit of less overhead for instructors. Whether there are 150 students, like the groups compared in the Digital Logic Engineering Course, or thousands of students, there is no additional overhead for normal lecture sessions in terms of requiring larger lecture halls.

Group Discussion and Tutoring

If the concept and example modules are not enough to answer student questions, additional help is available. Depending on the course, additional problem solving modules can be added in the form of additional online videos, posted sample problems and solutions, or using online tools like message boards, email lists, and online conferencing to provide extra help for students. Discussion can be done in the traditional in-class method and focus on different basic or harder topics and concepts, depending on the need of the participating students. These group discussions can be limited to 15-20 students and custom tailored for each group of students to specialized discussions (problem solving, concept explanation, prerequisite knowledge, large topic scopes, etc.). Utilizing online tools like the message boards or email lists, students can sign up for specialized group discussion sections to match their current needs for help. Discussions can be held on campus, off campus, or even online through conferencing software.

If office hours are not practical because of a very high instructor/teaching assistant to student ratio or limited college budget, in-person or online tutoring can still be provided either by the university or a third party source. Adding tutoring for students who are having difficulty with the course helps individualize the experience and keeps students on track with the rest of the class, allowing discussion of harder concepts in lecture, discussion, or online possible.

Topic Review

Each set of modules and homework is considered a topic milestone. At the end of each topic, the instructor or teaching assistant contacts each student to review their current progress. This typically involves feedback for their homework solutions, guides them in the correct direction for what materials to review or revisit, and makes sure they are still on track with the course.
The previously discussed modules all help students learn concepts and complete homework quicker than they would compared to a typical in-class methodology. Based on our experiments in [8] and the student feedback received, the majority of students did not keep up with course material and did not ask advanced questions when simulating an online course that used a typical in-class methodology.

The biggest factor seemed to be lack of self-motivation from students. To help reinforce their learning experience and keep them motivated to keep up with the course, we use the concept of the “Hawthorne effect,” in which subjects were more productive when they had a manager watching them work [23].It would not be practical to physically observe individual students did their work, like in Roethlisberger's original findings of worker management at Hawthorne Works in the 1930's [26], especially in a large distance learning course. Instead, 5-10 minutes is spent following up with students after each homework module, either via email or through comments listed next to student scores after grading. This type of milestone review can guide students and provide feedback on their current weaknesses and strengths.

Comments are more detailed than typical homework grading: students are given some direction on the concept they missed and suggested to watch certain videos or complete different practice problems if they had difficulty with certain ideas. The time available to spend on this module is dependent upon the availability of the instructor and ability to hire additional assistants to cover the class size.

Design Tool Metrics

To address the metrics measuring student progress based on the varying skill levels of individual students, each module of the design tool will be developed toward fulfilling each of the metrics for each level of students by analyzing how each module fulfills each groups' higher and lower order needs. These needs are adapted from Maslow's Hierarchy of Needs Theory (Figure 5), which identifies a pyramid of needs [27] [28].

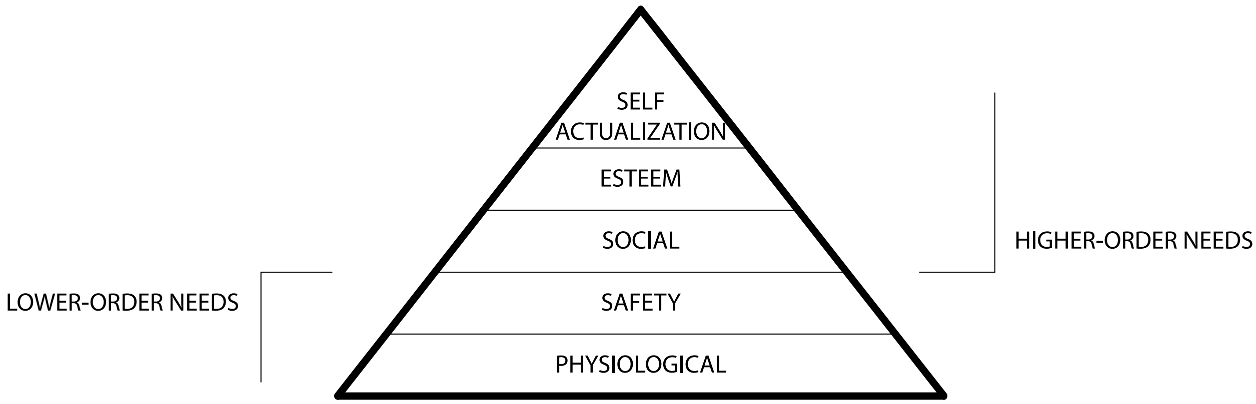

Figure 5: Maslow's Need Hierarchy Model adapted from [27]

Lower-order Needs

Physiological and Safety needs are lower-order basic needs. For a general situation, this typically refers to having housing, food, clothes, and basic necessities [27]. For students, this includes having convenient access to course content and related resources. Having access to content for the course, knowing expectations, and having simple questions answered are fundamental for any course. Furthermore, students need to be able to know their current grade standing throughout the offering of the course to help themselves assess whether or not their current strategies are producing the desired results. Inconsistent release of material and grades affect their basic needs and limit their motivation, performance, and satisfaction with the course. When these basic methods of communication of information is not readily available for students, such as situations where the class size is so large that grading takes longer than normative times, students become more anxious. This can lead to a negative self-perception and lower motivation. A negative self-perception of their own abilities will lead to subpar performance [21].

The automated advisor system focuses on these lower-order needs by providing ease of distributing and managing of course content for developers and present material to students in a timely and organized manner all available on a single course web site. More importantly, immediate feedback is available through self-check quizzes, which provide follow up recommendations based on students' performances. Current and previous works do not provide detailed recommendations based on individual students, if at all.

These needs are important for all levels of students, but are especially important for the beginner and intermediate students. By making sure these needs are met, student motivation and satisfaction can be maintained and even increased, allowing the rest of the design tool modules to improve student performance. Students in the digital design courses reviewed here respond positively to feedback on their activities, such as homework, as shown in the Appendix. However, as class sizes continue to grow, manual detailed feedback may not be practical. An automated system can provide the same basic feedback as a grader or TA in a fraction of the time, saving time for both educators as well as students.

Higher-order Needs

Social and Esteem needs are higher-order needs. In a general setting, people desire social interaction and affirmation from their peers and mentors [27]. Positive affirmation can help motivate people to continue doing the work they are doing, or even expanding on their current work striving to be better. Students also need interaction, a feeling of inclusion, group work, and constructive feedback. In terms of esteem, students need to build and maintain confidence and feel senses of achievement throughout the course, as well as get recognition when they perform well.

The automated education system provides an interface for developers to set goals and keep track of student achievements through generated reports. The tool will also provide an area for students to interact with each other. Feedback can be given to individual students on an automated basis to minimize the amount of time developers have to invest in the tool and course to keep these needs satisfied. Feedback from mentors is highly desired but also often hard to accomplish due to time restrictions. Many educators typically are over burdened with catering to lower-order needs, which leaves little to no time for higher-order needs. By automating many lower-order needs tasks, instructors have more free time to allocate to meeting with students and providing feedback. The automated education system helps facilitate meetings with students through student performance profiles, which help educators assess individual student's performance more quickly than manually looking up student grades for individual questions or requiring lengthy conversations each meeting to recall where the student's strengths and weaknesses are since there can be many students of whom individual instructors and TAs to keep track. Automated reports can help save significant amounts of time during a course offering.

The highest level of need, self-actualization, mainly affects advanced students. Students want to fully utilize their skills and abilities with challenging problems and grow as engineers. Optional modules and additional advanced material can be customized by developers in the design tool to facilitate this need for students who need more challenge. Satisfying this level of need is typically accomplished from in-person interaction. Freeing up available time from satisfying simpler needs first allows the user to utilize more time focused on higher level interactions in person. For the time being, this level of need is not addressed directly by the automated education system. Instead, the system frees up time for the instructor to pursue this area manually by automating other tasks to make this area manageable.

Methodology Comparisons

From [8], we can review some preliminary results comparing in-class, online, and hybrid approaches for the Digital Logic Engineering course (Table 1). Group A, consisting of 125 students, was used as the control group and was taught with an in-class methodology throughout the course. Group B, consisting of 142 students, was taught using an in-class methodology for Exam 1, an online methodology for Exam 2, and a hybrid methodology as presented in this paper for Exam 3.

Table 1: Group A vs. Group B Performance [8].

Exam 1:	Group A	Group B	Difference	Improvement
All Students:	81.31%	75.07%	-6.24%	n/a
Top 1/3:	90.24%	85.76%	-4.47%	n/a
Middle 1/3:	80.63%	74.58%	-6.05%	n/a
Bottom 1/3:	72.85%	64.86%	-7.98%	n/a
Exam 2:	**Group A**	**Group B**	**Difference**	**Improvement**
All Students:	65.28%	60.42%	-4.86%	1.37%
Top 1/3:	79.37%	79.44%	0.08%	4.55%
Middle 1/3:	64.92%	60.14%	-4.78%	1.27%
Bottom 1/3:	51.22%	41.67%	-9.55%	-1.57%
Exam 3:	**Group A**	**Group B**	**Difference**	**Improvement**
All Students:	53.52%	54.72%	1.20%	7.44%
Top 1/3:	70.77%	72.53%	1.76%	6.24%
Middle 1/3:	54.23%	55.31%	1.09%	7.14%
Bottom 1/3:	35.12%	36.32%	1.20%	9.18%

The top 1/3 of students actually perform better online than in-class, which is expected since they are allowed to proceed at their own pace and can cover more material than a typical in-class session; however, the bottom 2/3 of students do not perform as well. These students typically procrastinated in watching videos and doing homework, as evident from survey feedback and timestamps on their electronic homework submissions.

In the hybrid methodology, students benefit as a whole and by various performance levels. The top 1/3 still benefit more than in-class and online methods. The middle and bottom 1/3 benefit much more. The main difference between the hybrid methodology vs. in-class and online methodologies contributing to this performance difference are the modules described previously encouraging students to stay on track and provide the additional help needed that is lacking in current in-class and online methodologies.

The in-class methodology is not successful in keeping students on track, lowering performance of the various groups from about 20-40% by the end of the course. There is some performance decrease expected since the topics get more complex while design and homework problems become more involved as the course proceeds; however, 20-40% is quite a lot. Overall, using in-class, online, and finally hybrid, we see that performance only decreases by 15-30% by the end of the course. We expect this number to be less if the entire course is taught in a hybrid method.

As preliminary testing (Table 2), we compare Group A from [8] with Group C, consisting of 46 students, from a Summer 2013 Digital Logic Engineering Course, which received hybrid instruction exclusively for the entire duration of the course utilizing the hybrid methodology reviewed earlier.

By sticking to the hybrid methodology, students are on track from the beginning, as seen by the much smaller difference between the highest and lowest performing groups. Since Group C is caught up at the beginning, their performance remains relatively higher than Group A throughout the course. Improvements are seen across the board in all exams and groups comparing the hybrid solution to the in-class methodology.

Table 2: Group A vs. Group C Performance.

Exam 1:	Group A	Group C	Difference	Improvement
All Students:	81.31%	93.60%	+12.29%	n/a
Top 1/3:	90.24%	98.38%	+8.14%	n/a
Middle 1/3:	80.63%	97.05%	+16.42%	n/a
Bottom 1/3:	72.85%	91.61%	+18.76%	n/a
Exam 2:	**Group A**	**Group C**	**Difference**	**Improvement**
All Students:	65.28%	87.55%	+22.27%	+9.98%
Top 1/3:	79.37%	96.48%	+17.11%	+8.97%
Middle 1/3:	64.92%	98.95%	+34.03%	+17.61%
Bottom 1/3:	51.22%	74.55%	+23.33%	+4.57%
Exam 3:	**Group A**	**Group C**	**Difference**	**Improvement**
All Students:	53.52%	83.05%	+29.53%	+17.24%
Top 1/3:	70.77%	93.42%	+22.65%	+14.51%
Middle 1/3:	54.23%	84.97%	+30.74%	+14.32%
Bottom 1/3:	35.12%	71.53%	+36.41%	+17.65%

The automated education system developed for this research focuses on following a hybrid methodology. Each feature provided by the tool aims to facilitate increased student motivation through concise and time feedback with the primary metric of minimize waiting time for students to get feedback and lookup/search times for instructors to find references for individual students. By automating many of these lower-order and simpler needs from students, the aggregate time saved by both students and instructors throughout the offering of a course is available for pursuing higher level and more difficult concepts, which provides more satisfaction and quality of education for all participants. As more tasks are automated, larger class sizes can be supported with minimal increase in overhead.

Chapter 4: Active Learning Personal Advisor Tool Overview

The automated education system, the Active Learning Personal Advisor, primarily focuses on improving the following metrics: student performance, motivation, and course satisfaction. Students are split into several groups based on their skill and performance level. This tool has a front-end for students and a back-end for instructors and teaching assistants. This design methodology is unique to our design tool and is not currently implemented by other tools managing MOOC's.

Front-End Interface Overview

The front-end interface contains modules and tools for students. Each of these modules has a corresponding interface to allow instructors to populate each module with relevant data for the design tool to generate a student profile, analyze the profile, and make suggestions to students. These modules help satisfy basic student needs of the course under Maslow's Hierarchy of Needs Theory. For each section below, the examples provided will be various student performance scenarios for a Digital Design 101 course where the student is progressing through the RTL Combinatorial Components portion of the course.

Concept and Example Modules

Concept and example modules replace the primary lecture and discussion sessions found in traditional classrooms. Each concept and example module is approximately 5-20 minutes in length. Students watch videos and answer questions while instructors create videos and provide questions for concept checking.

Concept and Example Modules Student Point-of-View

Students are presented with a list of concept and example modules. A brief quiz is provided at the end of each module. Students answer these questions and depending upon their performance, the design tool will analyze their answers and make suggestions for additional studying.

For example, a student may have just finished viewing the Arithmetic concept module. At this point, some review questions will be prompted to the student, such as:

For a 4-bit Two's Complement Adder/Subtractor, what is the result of 0010 - 1101? ____

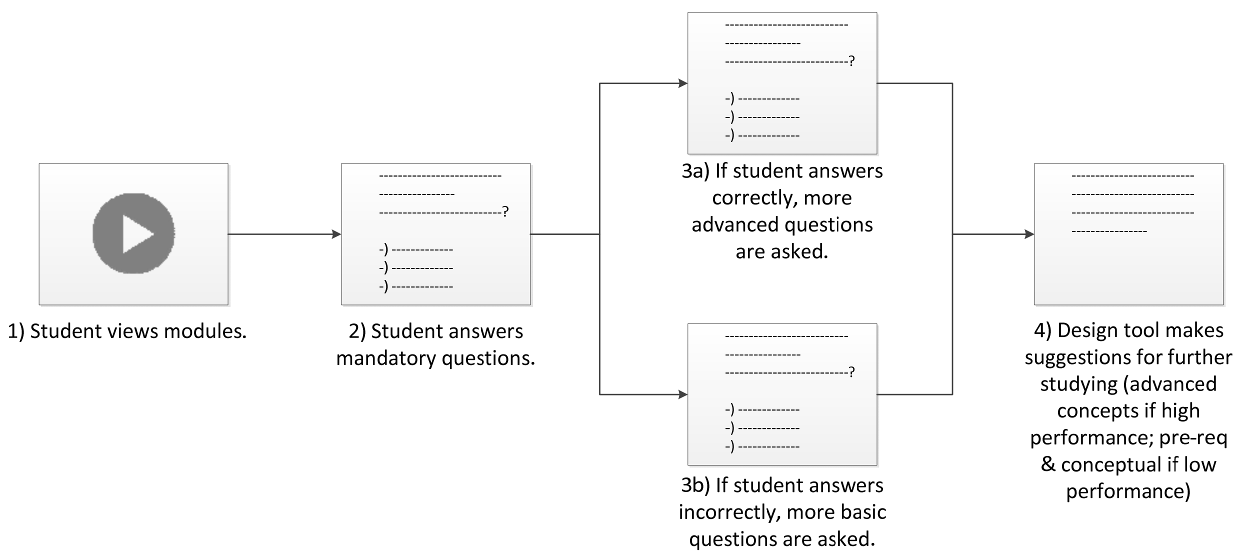

Figure 6: Concept Module Flow

If the student answers correctly with 0101, a more advanced question can be asked to figure out the understanding of the student. If the student answers incorrectly with a wrong answer, more basic questions can be asked to assess the student's level of understanding. The design tool will then compare the keywords associated with the questions against the concept, example, and pre-requisite modules. Based on the answers, the tool suggests further studying material. For example, if the student answered fundamental questions incorrectly, it may determine that they need to review basic binary arithmetic before coming back to two's complement numbers. If the student answered all questions correctly, it may be suggested that the student review more advanced concepts related to adders/subtractors to keep them challenged.

Concept and Example Modules Instructor Point-of-View

Instructors upload videos they have created for each concept and example in the course. For each of these modules, the instructor can provide additional data such as keywords of main concepts covered, as well as the time range and slide numbers which they are covered. Instructors decide which concept and example modules are shown as the primary lecture material and which modules are only for follow-up help with specific concepts.

Along with each module, quiz and self-check questions can be inputted. The instructor will also provide additional keywords for each question asked to the design tool. Each question can be filtered to be always asked after viewing a module or as follow-up questions for further studying.

For the question previously mentioned, the instructor can add keywords, such as "binary arithmetic", to help the design tool find proper modules when a student answers questions incorrectly.

Online Tools

Online tools such as message boards provide an environment for students and instructors to interact and discuss varying concepts regarding the course. These tools satisfy the social and esteem needs portion of Maslow's Hierarchy.

Online Tools Student Point-of-View

Threads and posts from the course message board can be added to the data set for each student profile by the design tool with help from instructors and teaching assistants (**Figure 4**). Keywords can get generated for each post by the design tool to further analyze the needs of students.

The general format will ask a student to classify their own post among several different options, such as "question", "clarification requested", "typo/error found", etc. Posters can also vote for the "best solution" when a question is posed as well.

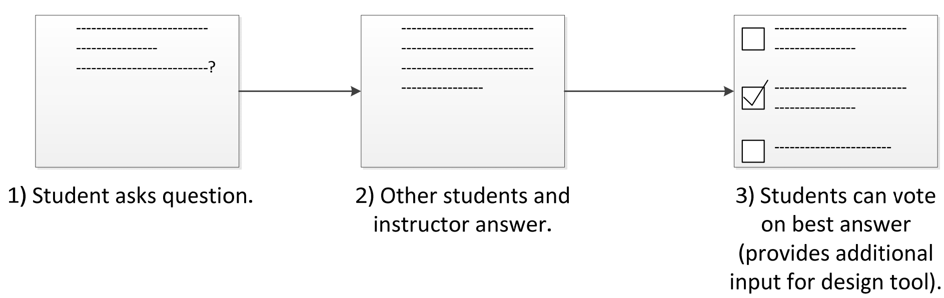

Figure 7: Online Tools Flow

Continuing with the RTL combinatorial module example with two's complement adders, a student may post a question regarding how to handle overflow in such cases. Other students, and possibly the assistants and instructor, will answer the post. Participants vote on the best answer. The current best answer is highlighted for the topic thread and is later used by the design tool when making studying suggestions.

Online Tools Instructor Point-of-View

Similar to most online message boards, instructors and teaching assistants can directly interact with students. Instructors can also vote for “best solution” posts in threads. To further help the design tool make meaningful interpretations of the posts and threads on the message boards, instructors and teaching assistants can add additional keywords for concepts covered by each thread. These keywords can be used as additional input for the design tool.

Topic Review

When homework and initial viewing of the concept and example modules are completed, the topic review can be automatically generated by the design tool, which will send suggestions for further studying to students based on their profile and data gathered from them for each topic. This automated topic review can be sent via email automatically, minimizes the need for instructor or assistant initial intervention, and maintains the students desire for “one-on-one” interaction.

Topic Review Student Point-of-View

Students receive their review via email after completing the required modules and homework. Homework is downloaded, completed, and submitted online through the design tool or equivalent online interface (**Figure 5**). The email will contain information regarding the student’s current progresses, his or her strengths and weaknesses, as well as suggestions for additional review to help the student catch up if needed. For advanced students, additional challenging modules are recommended.

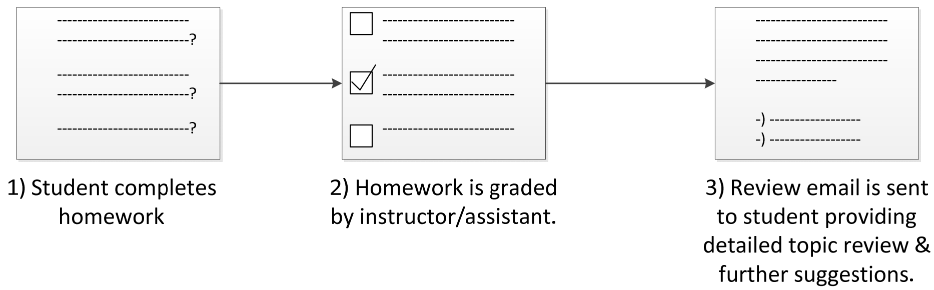

Figure 8: Topic Review Flow

The homework for RTL combinatorial components contains several concept questions. Each student’s homework is graded by the instructor and assistants. Optional comments may be left for the assignment. The design reviews the homework grades, matches it against previous suggestions, checks student module viewing behavior, and generates a detailed review email to be sent to each student. The review email can suggest more detailed review modules than the suggestions from the concept and example modules.

Topic Review Instructor Point-of-View

Instructors and assistants will add similar data for each homework question as they did for quiz questions. The data may contain keywords for concepts, difficulty of the question, and links to additional follow-up questions and modules. This information is only entered once per homework. Once a student’s homework is graded, the grade and any additional comments from the instructor or teaching assistant are added to the student's profile. From here, the design tool can generate an automated message for the student.

Group Discussion and Tutoring

Discussions and tutoring sessions are minimized since the design tool will be able to cater to most basic and intermediate concepts with which students are struggling based on the student profile developed from earlier.

The data gathered from the concept and example modules and quizzes, homework and topic review, and the online tools modules can be analyzed by the design tool through intelligent algorithms to provide additional basic and intermediate help for students by suggesting more modules to watch, different questions to attempt, and certain threads in which to review and participate.

When instructor or teaching assistant intervention is required, the design tool can provide suggestions to the instructor and assistant by presenting an overview of the student's strengths and weaknesses in order to make in-person or online meetings more efficient.

Back-End Data Structure and Algorithms Overview

The back-end of the design tool consists of the data structure to store content and information as well as the algorithms responsible for analyzing, assessing, and producing recommendations for the user. The primary concern for data structure requirements is to store content in a way that will allow ease of searching while minimizing overhead. The primary focuses of the algorithms utilized are to allow for quick response times within modules, such as the recommendation system for post-video concept module quizzes, while providing deeper and more thorough recommendations and analysis for topic review modules.

Data Structure Requirements

Both implementation time for scalability purposes as well as searching efficiency were considered for the back-end of this tool when choosing a data structure. Although a flat file database could be quickly implemented and take up minimal space, it would not be optimal for scalability for a larger number of courses and large number of users accessing the same resource. So, despite requiring more start-up effort and space, a database schema is more suitable handling multiple users and storing data in an accessible way for easier accessibility compared to a flat file [29] [30]. A completely custom data structure built specifically for this course implemented directly in the final programming language choice would allow for the fastest searching and analysis since data can be directly accessed without conversion or importing; however, it would require significantly more implementation time compared to storing course content by building a data structure using currently available database tools.

Therefore, in order to reduce overhead time required in creating a course data structure from scratch, the data structure created for the Digital Design 101 course was built in MySQL to maximize compatibility with existing open-source software. Many popular course management tools, such as WordPress and Moodle, utilize MySQL as their back-end data structure [31] [32]; therefore, the data structure that was chosen for this tool is also built using MySQL. Table and field choices are designed for modularity for this course and can be scaled for other courses, as well. Each record, or entity, in each table, or collection of entities, contains all relevant data needed by the algorithms accessing them. Records and tables are designed using best practices as described by [33] and [30] for building databases using the concept of Entity-Relationship data models.

Online content for students reaches approximately 4 GB of data for Digital Design 101. The majority of this data is comprised of online concept and example video modules. The basic database, which contains all questions, student answers, and assessments for the instructor to utilize, is approximately 22 MB for the 10 week course with 114 students participating in self-check quizzes. For this particular course, there are 152 self-check quiz questions, totaling approximately 2.5 MB of storage space. Student answers are recorded for future analysis, which occupied approximately 4 MB in the database. The database size grows to approximately 351 MB, which is still reasonable, when scaled to a course of 10,000 participants based on existing course metrics gathered from online self-check quizzes from the current course. From a storage perspective, supporting 10,000 students is feasible with the chosen data structure.

Course Data Structure

The course data structure (Figure 9) contains all course level specific data including but not limited to the following:

- Course Name/Number
- Concept and Example Module Information
    - Keywords for concepts covered
    - Whether it is included in the primary lecture materials or provided as an additional module to be used later
    - Annotations of timeframe for each concept covered
    - Slide references for each concept covered

- Additional Modules
    - Prerequisite reference modules, additional basic concept modules, step-by-step example modules, advanced concept and process modules
- Quiz Questions
    - Keywords for concepts covered
    - Whether it is included in the quiz or provided as an additional question to be used later
    - Difficulty of question (range from basic conceptual understanding to advanced process knowledge)
    - Specific links or references for question (others will be derived from the design tool itself)
- Message Board Posts/Threads
    - Categorized by types of questions asked (beginner, intermediate, advanced – filled out by instructor or assistant)
    - For each thread:
        - Keywords for concepts covered
        - If it was a question: whether question was fully answered, needs follow-up, or needs one-on-one meeting
        - if it was an answer: whether it fully answered the question, needs follow-up, or needs one-on-one meeting
    - For each post
        - Keywords for concepts covered
        - If it was a question: whether question was fully answered, needs follow-up, or needs one-on-one meeting
        - if it was an answer: whether it fully answered the question, needs follow-up, or needs one-on-one meeting
- Homework Questions
    - Keywords for concepts covered
    - Whether it is included in the homework or provided as an additional question to be used later
    - Difficulty of question (beginner, intermediate, and advanced)
    - Specific links or references for question (others will be derived from the design tool itself)

The above data is utilized by the design tool to generate automated reports and suggestions to students, namely after quiz or homework questions are answered.

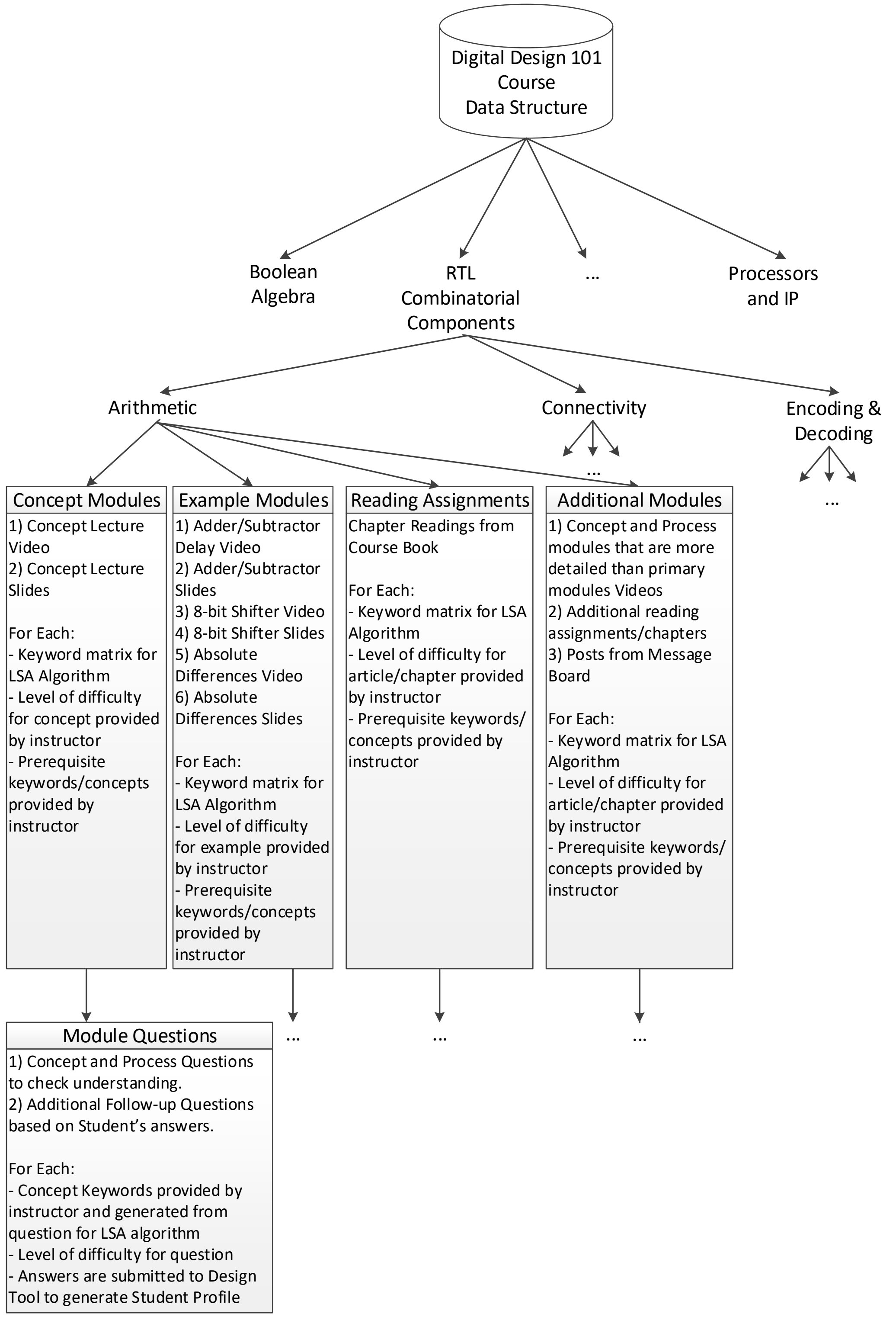

Figure 9: Partial Course Data Structure (Arithmetic portion of RTL Combinatorial Components)

Student Profile Data Structure

The student profile data structure (Figure 10) contains individual profiles for each student. Each profile contains the following information:

- Student Name/ID
- Student Level (beginner, intermediate, advanced)
- List containing number of times each module is watched
- Quiz Information
    - Questions answered correctly/incorrectly
- Message Board Information
    - Types of questions student asks
    - Types of answers student gives
- Homework Information
    - Questions answered correctly/incorrectly
- Summary of Strength & Weaknesses
    - Generated by design tool for instructor use and/or topic review for student

The above data is utilized by the design tool to make customized reports and suggestions to students based on the profile data and course data. This data is also used to generate student summaries for instructors to review and gather ideas for creating additional content.

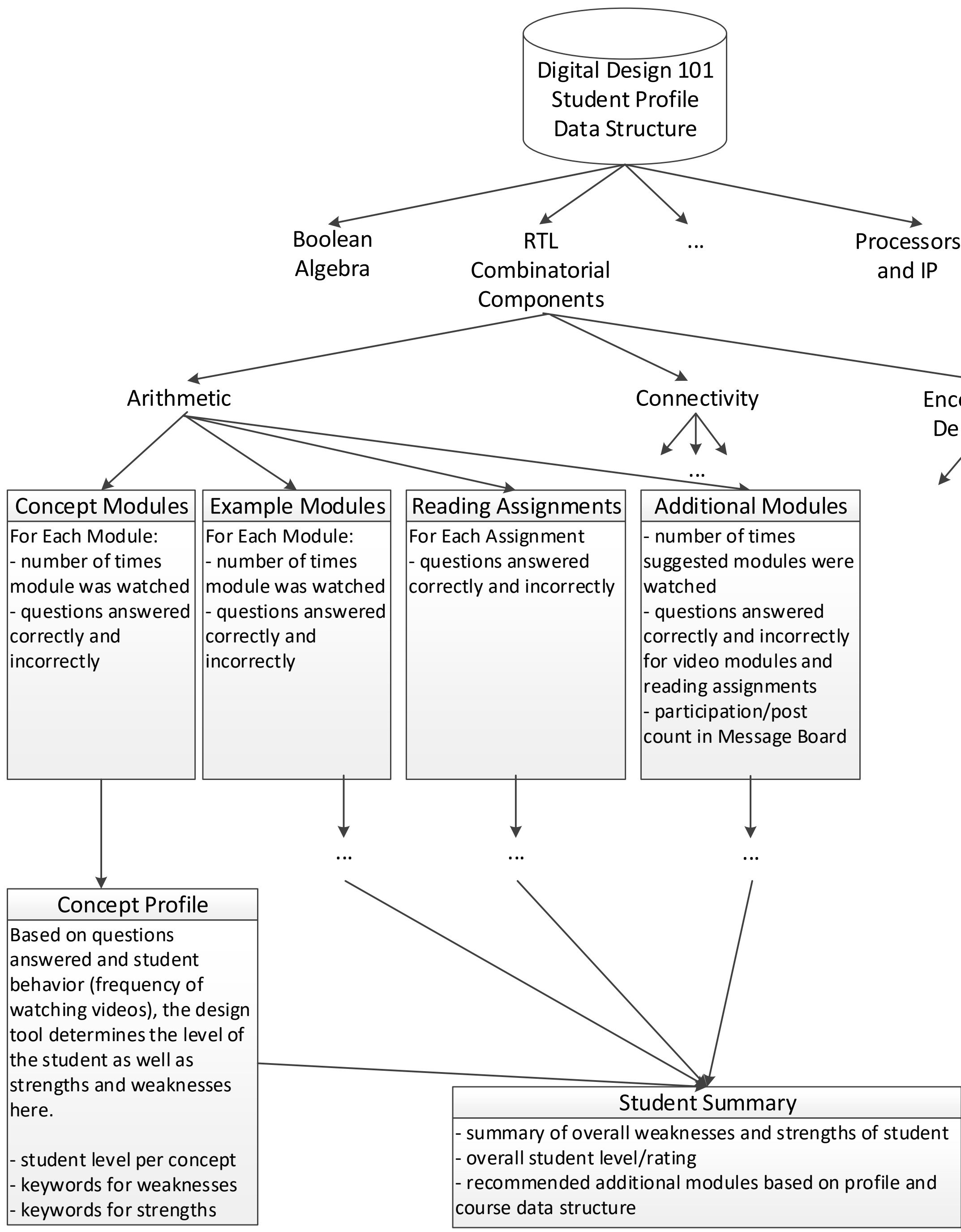

Figure 10: Partial Student Profile Data Structure

Design Tool Algorithms

The design tool analyzes data from the design tool data structure in order to create custom reports for students and assign them additional modules for further studying when appropriate. This tool also provides a summary of each student's progress for instructors to review before meeting with students individually when needed to maximize efficiency when holding office hours. This tool will search through a large amount of data, as shown in Chapter 5: Experiments and Results, but will restrict its searches to just course material provided versus to reduce information overload [34].

Student Feedback Generation

After students each concept and example module, students answer a series of questions to check their understanding of the material. Depending on their answers, the design tool will rate the students, look through the data structure for additional modules that will help them, and provide additional feedback as appropriate.

The design tool bases its keyword search (Figure 11) on a Latent Semantic Analysis (LSA) algorithm to find similar keywords and concepts based on the provided dataset. This can be further expanded by indexing message board posts and referencing the data inputted for each post by instructors and assistants.

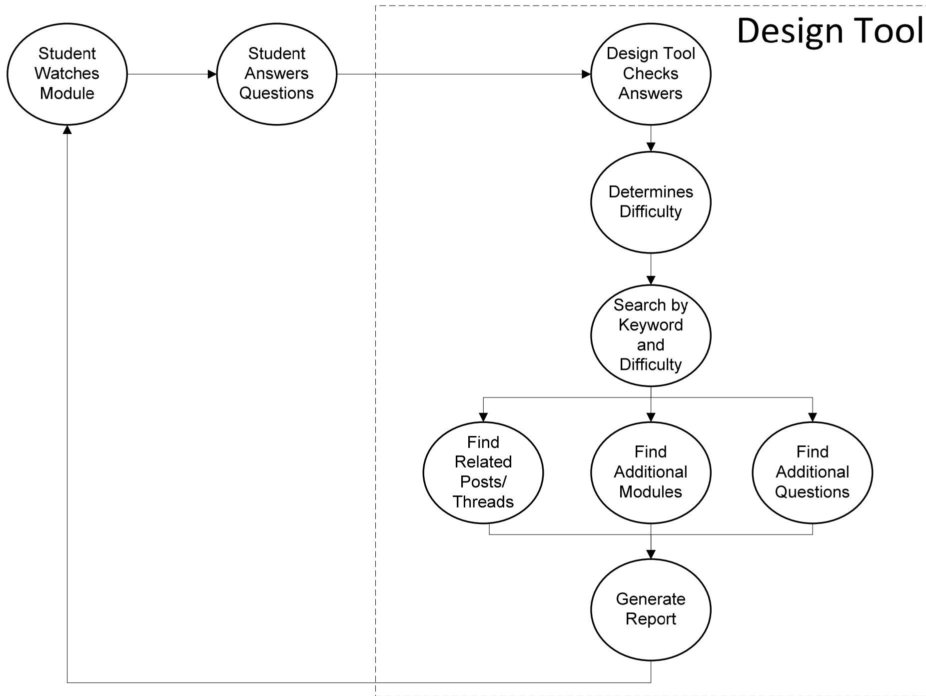

Figure 11: Student Feedback Generation

The LSA algorithm is used to find modules, posts, and questions based on the keywords of the quiz or homework questions the student misses. From there, the design tool determines which modules are most appropriate based on difficulty, closeness in relation, and previous topic review suggestion results.

Student Summary for Instructor Generation

Instructors will inevitably be required to meet with some students, be it in-person or online. Since the goal of the design tool is to allow for an individualized experience for students on a large scale, student summaries are generated by the design tool (Figure 12) to help instructors quickly identify a student's weaknesses and strengths before meeting with him or her.

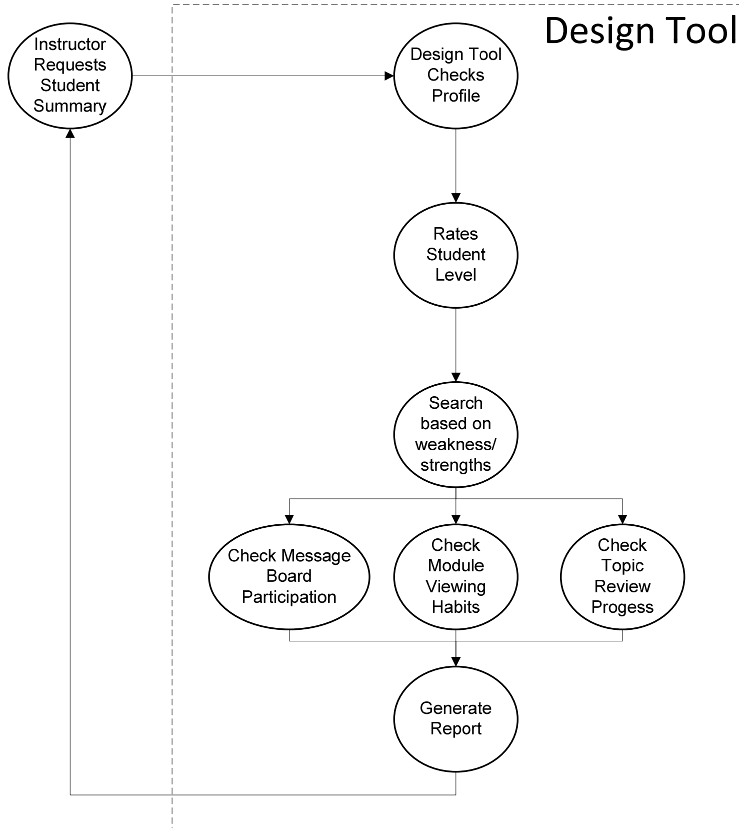

Figure 12: Student Summary Generation

Similar to the Student Feedback Generation, a Latent Semantic Analysis algorithm to find matches. The LSA algorithm is used to find modules, posts, and questions based on the keywords of the quiz or homework questions the student misses and gets correctly. The design tool further analyzes how the student divides his or her time among various concepts. Suggestions are made based on student participation and focus on the modules.

Basic Student Assessment based on Self-Check Quizzes

Basic student assessment is possible through techniques similar to standardized testing. A set of questions of varying difficulty are asked after each concept and example module. Depending on the student's answers, harder questions are asked if the student answers correctly and simpler questions are asked if the student answers the questions incorrectly. At a minimal, this information can be presented to the instructor as part of the student profile report. From this information, student can be rated on a scale of 1-5, with 1 representing students who are struggling with basic questions and 5 representing students who are answering most or all questions correctly. The granularity of student performance levels can be adjusted in the data structure by the instructor or course developer.

Table 3: Initial Student Assessment from Self-Check Quizzes.

Perf. Lvl	# of Students
1	68
2	39
3	168
4	81
5	232

For the table above, 588 students from the Digital Design courses participated in the online self-check quizzes. The design tool automatically rated the general performance level of each student. The students' performance levels are adjusted accordingly as the course progresses. For this particular course offering, the self-check quizzes were limited in difficulty since the priority was to make all self-check quizzes automatically graded by the education system. Therefore, even many of the more difficult questions did not involve as much critical thinking and time as a typical difficult question in a homework or exam assignment may involve. In future work, with additional difficult questions available that are automatically graded, a larger range of performance levels can be assessed.

Additionally, references can re-prioritize depending on metrics such as question difficulty, source reference difficulty level, and student performance. For the purposes of this design tool, these metrics are recorded but do not affect results since more source reference data is required. This is instead left as future work for further implementation.

Search and Analysis Algorithm Requirements

Different metrics have varying priorities when choosing search and analysis algorithms to perform tasks. The recommendation system for this tool has a mix of time-sensitive as well as non-time-sensitive tasks, so different implementations will be utilized for each type of task. For the time-sensitive tasks, such as responding to students with recommendations after they take a self-check quiz for a concept or example module, the response time needs to be considered. Internet users find response times under 3 seconds acceptable [35] up to several seconds depending on application [36]. The searches conducted in Chapter 5: Experiments and Results are completed in less than .1 seconds, so will not be a limiting factor for real-time responses if needed. Furthermore, many search results and recommendations can be sent after a specified time, say via a scheduled email, to allow the user time to process the material if desired by the instructor or course developer. For other tasks, such as study tips and advice sent to a student after a topic review or midterm, time is not as big of a factor, so the search space can be larger.

Beyond response time, scalability and modularity metrics in terms of allowing straight forward implementation across different courses should also be considered when choosing a type of algorithm to use. Although a straight-forward keyword matching algorithm would be able to provide some very basic recommendations, artificial intelligence and natural language processing algorithms have the potential to provide more in depth analysis and recommendations. A straight forward keyword search would be very fast, but it would be very limited in its ability to make complex computations or in-depth analysis of why a student may be struggling with a particular concept. However, any algorithms utilized should still be able to provide, at minimal, similar results as a traditional keyword search for basic queries.

Many artificial intelligence algorithms can be made to work very well because the specialized language and jargon of engineering can be built in; however, implementations will be specific per class, which will limit modularity of any product created. Although it may be useful to integrate these algorithms in future work, for the current proof-of-concept build, algorithms built specially to cater to a single course are omitted from implementation. The natural language processing algorithm choice for this tool is latent semantic analysis (LSA). LSA is chosen because it allows for a more universal approach since there is no inherent need for specialized understanding of any keywords or extensive knowledge of material covered in relation to comparable algorithms and methodologies [37] [38] .

Latent Semantic Analysis Overview and Implementation

Rooted in psychology and math, LSA categorizes and organizes content for the search space based on multiple matrices that correlate keyword frequency and distance through vectors. Ultimately, LSA can be used to map similarities between different content (in this case, videos, slides, chapter sections, and online references) based on different metrics. The algorithms for LSA are further re-purposed and utilized in the tool to come up with recommendations for students that are more sophisticated than a simple keyword match. It can also be used to generate an assessment of student performance to aid the instructor in analyzing the effectiveness of current course material and to provide further feedback for their students.

Although LSA has no contextual knowledge of the data through which it searches, domain knowledge and recommendations can be equivalent to what both experts and novices may recommend based on reading a large number of documents [39]. In this work, LSA predictions closely correlated with that of the human subjects. In cases where LSA did not predict correctly, the underlying factor was attributed to not having enough data to search [38] [39]. For this design tool, results are initially normalized based on what an instructor or TA may expect to recommend based on the given inquiries. Results are further compared to actual recommendations from TA's and instructors through email, homework comments, or in-person feedback during office hours or lectures and discussions.

LSA Implementation

First, for latent semantic analysis to work, a data structure containing all of the content to be searched must be provided. For the Digital Design course, the design tool will be searching primarily through 2 of course books [40] [41], online concept and example video transcripts, and slides. This content can be transcribed manually or parsed automatically by a program if there is a digital version available. For the purposes of this course, content was entered manually for the processing portion of the system.

The content is processed by removing stop words and entered into a custom data structure, described in the previous section: Back-End Data Structure and Algorithms Overview. Stop words consist of articles, conjunctions, and other common words that may not be as useful as nouns. Removing these words lowers the amount of storage required in the data structure, minimizes the noise in the dataset, and speeds up the search algorithms since there is less data to consider. The initial context/relationship matrix contains keywords used throughout the sources and the documents in which they are present, as well as how many times they appear in each document. A partial view of the relationship matrix is shown in Figure 13.

	Introductic	Introductic	Introductic	Introductic	Introductic	Case Studi	Case Studi	Case Studi	Case Studi	Case Studi	Case Studi
arithmetic	0	0	0	0	0	0	0	0	0	0	0
arm	0	0	0	0	0	0	0	0	0	0	0
array	0	0	0	0	0	0	0	4	0	3	0
arraystyle	0	0	0	0	0	0	0	0	0	0	0
arrive	0	0	0	0	0	0	0	1	0	0	0
arrow	0	1	0	0	0	0	0	0	0	0	0
art	0	0	0	0	0	0	0	0	0	0	0
ascii	0	0	0	0	0	0	0	0	0	0	0
asm	0	0	0	0	0	0	0	0	0	0	0
aspect	0	0	0	0	0	1	0	0	0	0	0
assembled	0	0	0	0	0	0	0	0	0	0	0
assembly	0	0	0	0	0	0	0	0	0	0	0
assert	0	0	0	0	0	0	0	0	0	0	0
asserted	0	0	0	0	0	0	0	0	0	0	0
asserting	0	0	0	0	0	0	0	0	0	0	0
assign	0	0	0	0	0	0	0	0	0	0	0
assigned	0	0	0	0	0	0	0	0	1	0	0
assigning	0	0	0	0	0	0	0	0	1	0	0
assignmen	0	0	0	0	0	0	0	0	0	0	0
assist	0	0	0	0	0	0	0	0	0	0	0
assistance	0	0	0	0	0	0	0	0	0	0	0
assistant	0	0	0	0	0	0	0	0	0	0	0
associated	0	0	0	0	0	0	0	1	2	1	2
associatin	0	0	0	0	0	0	0	0	0	0	0

Figure 13 : A partial view of the initial context matrix.

For these relationship matrices, each document was separated by the headings according to their books/sources. For example, the introductory chapter of [40] contains 5 subchapters with an additional 11 separate subsections. The sources can be split up by chapter, subchapters, subsections, or a combination of any of these. For the purposes of creating more detailed recommendations, the sources for the digital design course were split up primarily by subsections. From Figure 13, the partial view of the headings shows several different Introduction chapter references, which point to specific subchapters and subsections. As an example, a recommendation of “Chapter 1.2 The World of Digital Systems – Converting from Decimal to Binary Using the Addition Method” is typically more useful than just “Chapter 1” or “Chapter 1.2 The World of Digital Systems” [40].

However, Figure 13 only represents the initial matrix. For any induction or inferencing to occur, the data structure must be modified. In particular, singular value decomposition is performed, and the diagonal matrix that represents the number of dimensions of the dataset is reduced. The lower the number of dimensions, the more induction is typically conducted [37]. Figure 14 shows a partial representation of the data structure once dimensions are lowered to 64 from over 440.

	Introductic	Introductic	Introductic	Introductic	Introductic	Case Studi	Case Studi	Case Studi	Case Studi	Case Studi	Case Studi
arithmetic	0.06	0	0.03	-0.15	-0.11	-0.08	0.03	0	0.08	-0.09	0.12
arm	0	0	0.03	-0.22	-0.08	0.03	0.09	-0.39	-0.02	-0.21	-0.12
array	0.27	2.17	4	0.13	-6.48	3.24	2.12	-0.22	-0.39	4.19	-15.31
arraystyle	0.15	0	0.02	0.02	-0.08	-0.11	0.04	0.04	-0.07	-0.04	-0.02
arrive	-1.31	0.01	0	1.49	0.66	-1.38	1.37	0.6	-0.18	0.88	0.14
arrow	-0.86	-0.67	-0.21	1.24	0.51	-0.44	0.39	0.29	-0.03	0.29	-0.19
art	-0.03	-0.03	0.06	-0.08	-0.02	-0.06	0.08	-0.22	0.06	-0.15	-0.03
ascii	-0.02	0.18	-0.04	0.07	-0.23	-0.02	0.08	-0.21	0.06	-0.18	-0.27
asm	-0.27	0.73	0.03	0.22	-0.58	-0.34	0.35	0.52	-0.36	-0.2	-1.3
aspect	-0.16	0.08	0.12	-0.11	-0.44	-0.2	0.39	-0.59	0.4	-0.66	-0.42
assembled	-0.01	-0.02	0.02	-0.05	-0.01	-0.03	0.04	-0.12	0.02	-0.08	-0.03
assembly	0.02	-0.03	0.05	-0.13	-0.07	-0.07	0.07	-0.19	-0.08	-0.1	-0.01
assert	0.05	0.79	-0.15	0	-0.11	0.02	-0.05	0.02	0.05	0.26	-0.71
asserted	-0.14	-0.64	-0.34	-0.11	-1.34	5.18	-2.94	2.34	1.78	-1.07	-1.38
asserting	-0.09	1.16	-0.54	0.13	0.37	0.72	-0.5	0.67	0.52	-0.42	0.04
assign	0.16	1.71	-0.54	0.4	0.06	-0.56	0.4	0.17	0.74	-0.19	-0.49
assigned	-0.61	2.38	-0.06	0.55	-0.37	-0.91	-0.08	0.02	-0.57	-0.38	-1.18
assigning	-0.29	0.61	0.55	0.22	-0.11	-0.4	-0.38	-0.16	-0.04	-0.17	-0.29
assignmen	-0.16	1.04	-0.18	-0.12	-0.11	-0.44	0.47	0.05	-0.61	-0.3	-0.43
assist	0.03	-0.02	-0.05	-0.03	-0.05	-0.01	0.02	-0.27	0.39	0.53	-0.01
assistance	-0.01	-0.07	0.06	0	-0.1	-0.12	0.07	-0.1	-0.1	-0.05	0.02
assistant	-0.02	-0.01	0.01	-0.04	0	-0.01	0.05	-0.16	0.05	-0.06	-0.02
associated	-0.9	0.9	2.14	0.53	0.13	-0.1	-0.12	-0.01	-0.22	-0.1	0.13
associatin	0.03	0.05	0	0	0	-0.07	0.07	-0.03	0.01	-0.03	0

Figure 14 : A partial view of reduced dimensions relationship matrix after SVD is applied.

As seen above, in Figure 14, values representing the potential relationship between keywords and documents in the SVD reduced matrix have been updated by reducing the number of dimensions for the matrix. Higher positive values indicate a closer relationship whereas lower and negative values indicate a lesser correlation. These updated values now allow us to search through documents that may not have certain words available at all.

A general pseudocode construction of the initial LSA matrix shown in Figure 15. This portion of the Active Learning Personal Advisor is implemented in the Python language. The LSA class is initialized with all necessary starting data structure, primarily arrays and dictionaries. From there, the stop word list is loaded into memory. After that is complete, all documents are loaded into memory for processing. This process only needs to be done once per document per update. The processed documents can be saved back to hard drive storage and read directly at a later time to avoid unnecessary reprocessing of stop words.

Once processing is complete, the initial matrix, similar to what is shown in Figure 13, can be created by processing each document and entering it into the LSA class data structure containing information of each keyword and how many times it occurs per source. The number of dimensions used for the initial matrix can be set specifically by the instructor and course developer. 64 dimensions were chosen for the digital design course through empirical testing of results returned from the LSA searches based on expected and past results the instructors and TA's have left for students via email, homework feedback, and office hour responses. This is also just the initial matrix. More matrices can be created based on user feedback for different levels of induction. Keywords and sources can be mapped into a concept space where keywords and documents that are closely related are physically located closer to each other than to documents which they are not closely related. This essentially creates clusters of knowledge, mapping words and documents together in groups. The map for 64 dimensions cannot be shown, but a simpler example is shown later for illustrative purposes.

```
# initialize our LSA processor class
activeSearchDS = lsaDataStructure()
# load our stopword list
activeSearchDS.loadAndBuildStopWordList('svdBetaStopWordList.txt')
# read docs from file
activeSearchDS.buildFullDocDictFromFile('./svdInputARMCh01.txt')
...
activeSearchDS.buildFullDocDictFromFile('./svdInputVideos07-FFandFSMpartial.txt')
# put it into the initial filtered list (removed stop words)
for doc in activeSearchDS.fullDocDict.keys():
   activeSearchDS.parseDoc(activeSearchDS.fullDocDict[doc])
# create default SVD matrix based on initial reduced dimensions
activeSearchDS.reducedDimensions = 64
activeSearchDS.calcReduced(activeSearchDS.reducedDimensions)
activeSearchDS.calcTermDict(activeSearchDS.reducedDimensions)
activeSearchDS.calcDocDict(activeSearchDS.reducedDimensions)
# read questions supplied by instructor/TA/website/data structure
activeSearchDS.readQuestionsFromFile('./svdQuestions02a.txt')
for question in activeSearchDS.questions:
   activeSearchDS.queryArray = question.split()
   recommendationList = activeSearchDS.processQuery(activeSearchDS.queryArray,
                                        activeSearchDS.reducedDimensions)
   # recommendationList can be displayed on screen, sent via email, or returned to the calling function
```

Figure 15: Pseudocode representation of inital LSA matrix creation

The relationship matrix is process through a method called Singular Value Decomposition, which allows the design tool to adjust the number of dimensions in which to process and search the data structure. If the matrix above represents Matrix A, then SVD is used to create 3 matrices, say W, S, and P, as follows:

$$\{A\} = \{U\}\{S\}\{V\}^T$$

Matrix U becomes the left singular vectors representing. Matrix S has singular values through its diagonal, and Matrix V^T represents the right singular vectors [42]. Matrix S controls the number of dimensions LSA utilizes to conduct its searches and induction through reducing the matrix size, and therefore reducing the dimensions used.

Matrix U becomes used as row vectors, representing the concept space for keywords and can be mapped into a concept space by reviewing the values in each row vector. Matrix V^T represents documents and sources concept space through column vectors, and can also be mapped onto a vector space, as well.

Documents and keywords may be mapped onto the same vector space, with documents that are closely related located closer to each other while documents that are not as closely related are in different areas, creating clusters as more documents and keywords are added [43].

Once the SVD matrices are created, questions can be retrieved by the instructor, TA, course web site, or data structure directly. For example, the course web site can call the search function of the automated education system after a student participates in a self-check quiz by supplying questions the students may have missed. These questions are read and processed similar to the sources. Each question or query can be mapped onto the same concept space as the keywords and documents. The documents or sources closest in cosine distance are considered more closely related to each other than sources that are further away in cosine distance.

To perform searches through this vector space, a query, say query q, can be represented by computing the centroid of vectors from the individual terms. From there, the cosine distance of each of the document vectors, say d_i, is calculated in relation to the centroid vector of the query and ranked from 0 to 1, with 1 being the closest match.

$$similarity(q, d_i) = \frac{d_i \cdot q}{|d_i||q|}$$

After the cosine distances from all documents to the query are processed, they can be ranked and displayed to the user. The experiments and results from these searches are presented in Chapter 5: Experiments and Results.

SVD Matrix Example

Since a 64 dimension matrix with more than 440 sources and over 3200 keywords is not practical to show in this dissertation, a simpler subset is presented here for illustrative purposes. In Table 4, the initial A matrix is populated with 7 sources and 13 shared keywords.

Table 4: Example initial A matrix representing LSA word frequency.

	c1	e1	e2	s1	s2	s3	t1
RTL	1	0	0	1	0	0	1
Combinatorial	1	0	0	1	0	0	0
Component	1	0	0	1	0	0	1
Arithmetic	1	0	0	1	0	0	0
Delay	0	0	1	0	0	1	0
Absolute	0	1	0	0	1	0	0
Differences	0	1	0	0	1	0	0
Digital	0	0	0	0	0	0	1
Design	0	0	0	0	0	0	1
Adder	0	0	1	0	0	1	0
Subtractor	0	0	1	0	0	1	1
Signed	0	0	0	0	0	0	1
Numbers	0	0	0	0	0	0	1

Legend:

c1: RTL Combinatorial Components – Part 1 (Arithmetic) Video

e1: Absolute Differences Video

e2: Delay in Adder/Subtractor

s1: RTL Combinatorial Components – Part 1 (Arithmetic) – slides 1-11

s2: Absolute Differences – slide 13

s3: Delay in Adder/Subtractor – slides 14-18

t1: Digital Design – Subtractors and Signed Numbers (Ch 4.6)

To keep the example simple, only titles are input into the example matrix as opposed to the full content of each source. After performing singular value decomposition, matrices U, S, and V^T are formed. These matrices are represented as the left singular matrix containing the keyword vectors in Table 5, the middle diagonal matrix containing information for setting dimensions of the LSA search space in Table 6, and the right singular matrix containing the document vectors in Table 7, respectively.

Table 5: Example U matrix representing keyword vectors.

RTL	0.52	-0.16	0.00	0.00	0.84	-0.04	-0.05
Combinatorial	0.32	-0.26	0.00	-0.35	-0.25	-0.67	0.44
Component	0.52	-0.16	0.00	0.00	-0.30	0.68	0.40
Arithmetic	0.32	-0.26	0.00	-0.35	-0.29	0.03	-0.79
Delay	0.09	0.47	0.00	-0.35	0.04	0.05	0.03
Absolute	0.00	0.00	-0.71	0.00	0.00	0.00	0.00
Differences	0.00	0.00	-0.71	0.00	0.00	0.00	0.00
Digital	0.20	0.10	0.00	0.35	-0.12	-0.14	-0.08
Design	0.20	0.10	0.00	0.35	-0.12	-0.14	-0.08
Adder	0.09	0.47	0.00	-0.35	0.04	0.05	0.03
Subtractor	0.29	0.57	0.00	0.00	-0.08	-0.09	-0.05
Signed	0.20	0.10	0.00	0.35	-0.12	-0.14	-0.08
Numbers	0.20	0.10	0.00	0.35	-0.12	-0.14	-0.08

Table 6: Example S matrix representing dimensions for searching.

3.25	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.54	0.00	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	2.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 7: Example V^T matrix representing document vectors.

c1	e1	e2	s1	s2	s3	t1
0.51	0.00	0.14	0.51	0.00	0.14	0.66
-0.33	0.00	0.60	-0.33	0.00	0.60	0.26
0.00	-0.71	0.00	0.00	-0.71	0.00	0.00
-0.35	0.00	-0.35	-0.35	0.00	-0.35	0.71
0.68	-0.08	0.18	-0.68	0.08	-0.18	0.00
-0.03	0.61	0.35	0.03	-0.61	-0.35	0.00
-0.20	-0.34	0.59	0.20	0.34	-0.59	0.00

For this example, this search space is reduced to 2 dimensions to show the updated relationship matrix. Reducing the dimensions and reassembling the matrix produces the A matrix in Table 8. From this matrix, the updated keyword and source relationships have been highlighted. Some keyword frequencies, such as "Combinatorial" and "Delay" have decreased in frequency for documents *c1*, *e2*, *s1*, and *s3*. Conversely, the same keyword frequency has increased for document *t1*. These relationships have been modified as a direct result of changing the number of dimensions used through SVD.

Table 8: Updated A matrix with 2 dimensions.

	c1	e1	e2	s1	s2	s3	t1
RTL	1.00	0.00	-0.01	1.00	0.00	-0.01	1.01
Combinatorial	0.75	0.00	-0.25	0.75	0.00	-0.25	0.51
Component	1.00	0.00	-0.01	1.00	0.00	-0.01	1.01
Arithmetic	0.75	0.00	-0.25	0.75	0.00	-0.25	0.51
Delay	-0.24	0.00	0.75	-0.24	0.00	0.75	0.50
Absolute	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Differences	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Digital	0.25	0.00	0.24	0.25	0.00	0.24	0.49
Design	0.25	0.00	0.24	0.25	0.00	0.24	0.49
Adder	-0.24	0.00	0.75	-0.24	0.00	0.75	0.50
Subtractor	0.00	0.00	1.00	0.00	0.00	1.00	1.00
Signed	0.25	0.00	0.24	0.25	0.00	0.24	0.49
Numbers	0.25	0.00	0.24	0.25	0.00	0.24	0.49

This relationship can be visualized through mapping the keywords and documents into a concept space based on the number of dimensions used. To accomplish this task, the automated education system takes the reduced U and V^T matrices and multiplies them with the reduced S matrix, producing each keyword's and document's cosine vector. The result of mapping these vectors is shown in Figure 16.

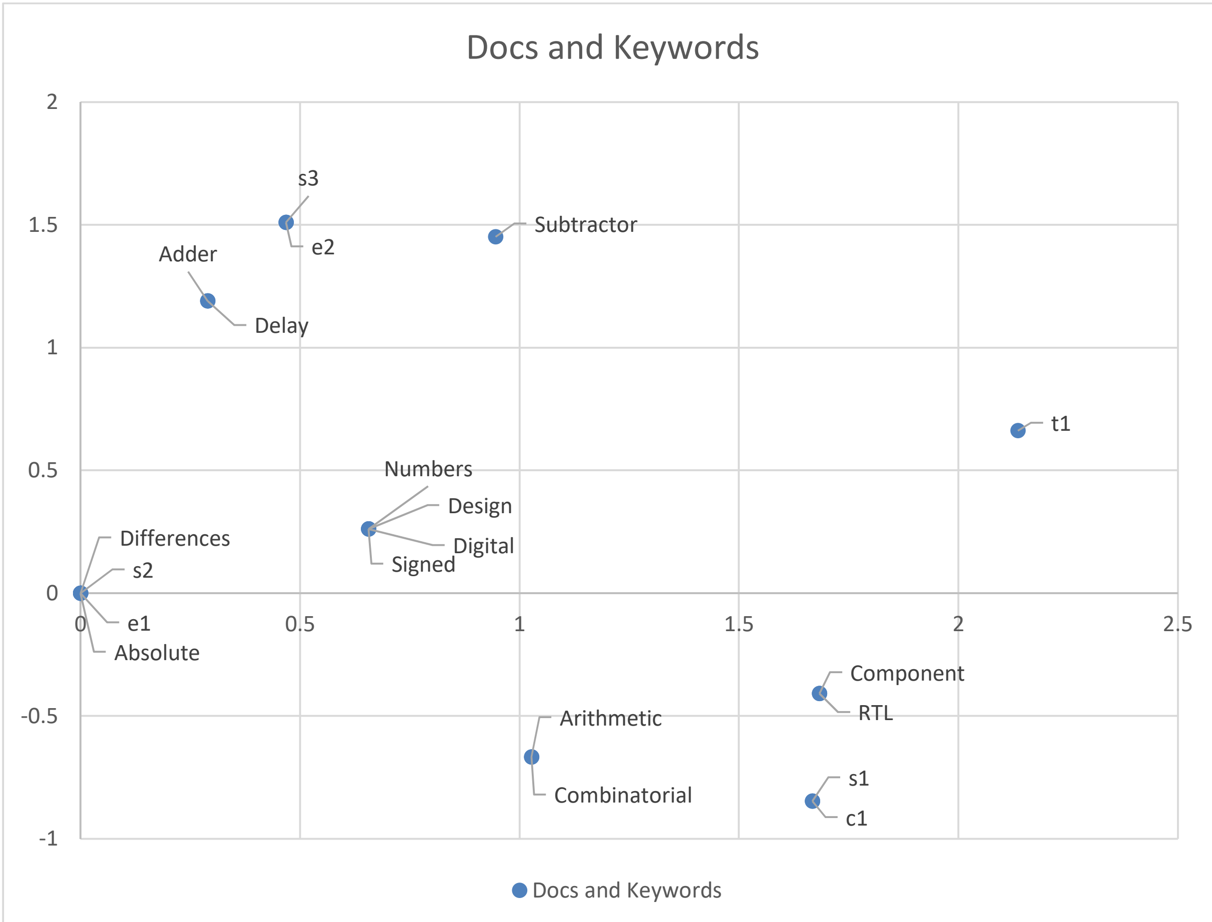

Figure 16: Example map of concept space for documents and keyword after SVD.

When a search is performed, the query is mapped onto the same concept space as the keywords and documents. Whichever documents are the closest to the query in terms of cosine distance are considered to be closely matched. Results are rated on a scale of 0 to 1, with 0 being not related at all and 1 being exactly related.

It is important to note that while many searches will return appropriate results, some search results for this particular example may not always provide the ideal matches. For example, searching for "Subtractor" will return documents *e2* and *s2* as the best matches while *t1* is not considered as close a match. Without context, this may be fine. However, if the user decides that *t1* should be a better match, the user can provide feedback letting the automated education system know that t1 should be a closer match. From there, different dimensions (3 and 4 in this case) can be utilized to created new reduced dimension matrices and reconfigured for future searches. If further steps are required, document *t1* can have its "Subtractor" keyword scaled through extra weighting of the keyword for that source. These weight values can be limited per user or performed system wide by the instructor or course developer.

LSA Limitations

LSA has some limitations, as well. Firstly, it does not understand synonymy or polysemy [37], which means it cannot quantify words that may have the same meaning. Fortunately, since the design tool is limiting its search depth to just potentially relevant course material, as decided by the instructor or course developer, the effect on the search results is minimal. As the design tool grows and expands, future implementations may need context knowledge; this is discussed in

Chapter 6: Future Work.

Speed and efficiency are potential bottlenecks and limitations of using latent semantic analysis [44]. However, with affordable contemporary high-speed processors, LSA performs much faster that it did just a couple decades ago. It is feasible to run LSA across large search spaces with modern computers. The data structure creation and search times performed as part of the experiments for the Active Learning Personal Advisor fall within acceptable user wait times [35], so specialized algorithm modifications are not required. That being said, additional improvements can be implemented for future implementations when the data structure becomes exponentially large.

Since LSA typically searches through a data structure that is constructed from simplifying text data to just keywords for easier information retrieval, known as a bag-of-words model [38] [45], some semantics is lost and ordering of words is not typically held. Again, since the design tool limits the search scope, and to some extent, the searches themselves because they'll be pre-emptively entered in via the design tool as opposed to a user for most cases, these issues are not a high priority. In a future implementation, steps can be taken to address the potential weaknesses of the data structure model by reintroducing some grammar so the order of words matter. For now, speed and scalability is the more desired outcome, so the data structure will maintain the typical model.

Matrix Search Space: Titles vs. Full Text

Latent Semantic Analysis can be used with just titles of documents or chapter headings or full text. In this section, we analyze the results of recommendations for various modules and compare these results when using titles or full document text to find the most effective results based on quality of recommendation, processing time, and storage requirements. Furthermore, the depth of matrix multiplications used for comparison can vary. A small number of dimensions provided reasonable results for the design tool when compared to actual feedback/recommendations from instructors and TA's for the Digital Design course. For full documents, the range of dimensions needed to produce adequate results can range anywhere from several dimensions to several hundred [46] [38]. Since the design tool only searches through several hundred sources for the purposes of these experiments versus the typical thousands of sources for more general searches, the number of dimensions required will be much less initially. The recommendation results based on varying depths and analyze their effectiveness over the various metrics used.

Storage Comparison of Titles vs. Full Text

Inputting full text, for example entire course books, into a data structure is more costly in terms of storage than just inputting the table of contents from a book. As a reference, the Table of Contents of the 2 course books used for experiments is under 75KB. The relevant course book chapters for the same course books are approximately 1MB. The transcripts from the online concept and example videos take up fewer than 350KB. Many courses will have varying amounts of content, but this is representative of a typical undergraduate 10-week engineering course. Although the textual information for a course like this could fit on an old floppy disk, students still often struggle locating the appropriate or relevant references – probably because the material itself normally consumes anywhere between 3-5 hours of reading/watching each week with just the first reading/viewing of the content. Creating the entire data structure with relationship matrices for all documents and keywords is under 90MB, which is still reasonable considering the amount of memory available in modern personal computers. Further optimization is possible through reducing, filtering, and combining more keywords.

For the purposes of this design tool, text is provided in full. There are also many common words, also referred to as “stop words”, which do not have an impact on search results, so they can be and are mostly removed. The design tool handles this upon reading the input data before including these words and sentences into the data structure. If input storage becomes a concern in the future (for example, if the design tool needed to search though all the books in a school library), preprocessing of “stop words” can easily be performed beforehand in order to minimize storage requirements. The current implementation leaves the input data alone until it is processed into the data structure to allow for adjustments, if needed, of certain stop words.

Lastly, since there is potential for customizing the LSA results per user by reconfiguring the data structure to be created with different search space dimensions, storage requirements will vary depending on final implementation decisions. These tasks can be done dynamically per search, which will save storage space but increase memory usage and vice versa with pre-computing several different matrices beforehand.

Speed Comparison for Results of Titles vs. Full Text using various Matrix Multiplication Depths

In terms of speed, creating a matrix of each table of contents (titles only), takes between 0.003 – 0.006 seconds. Creating a matrix for each book utilized in the search takes between 0.027 – 0.045 seconds. The most time consuming portion is filtering out stop words, or words that do not need to be part of the search, taking approximately 0.76 seconds for just titles and just under 48 seconds on average for the full text sources combined. Fortunately, this process only needs to be done once per source.

Constructing the initial co-occurrence matrix takes under 0.003 seconds and creating the Singular Value Decomposition matrices takes 0.121 seconds for titles only. For the full text sources, this process takes approximately 0.093 seconds and 1.59 seconds, respectively. Based on this information, constructing the SVD matrices and operating on them dynamically is possible, but it may be beneficial for user-response time if these matrices are built statically daily or weekly and stored as part of the data structure, depending on how often sources may need to be updated. Alternatively, the structures can be built dynamically if storage space becomes a concern.

Individual queries of the data structure take approximately 0.03 – 0.1 seconds depending on the length of the query and the size of the data structure.

LSA Comprehension of Text

Latent Semantic Analysis has no contextual knowledge of the dataset it is supplied. Its knowledge base is based on the psychological idea that similar documents would have similar keywords, layouts, and frequencies [38]. Since the design tool can restrict itself to searching only through potentially relevant sources, many of the short-comings of LSA are avoided. Additionally, search results are typically more relevant and granular since only domain specific knowledge is loaded into the data structure versus searching through a variety of potentially unrelated sources [39].

That being said, some contextual knowledge can be trained into the design tool through user feedback, both from students and instructors. If a student or instructor rates a certain result higher than another result from a list, keywords and documents can be weighted accordingly. Over time after enough user feedback, results should improve. The design tool data structure allows for weighting; however, weighting is not considered for the experiments and results in the following chapter and is instead left for future work since more data must be gathered.

Chapter 5: Experiments and Results

The following testing experiments are explored to analyze the effectiveness of using LSA as part of the design tool for generating recommendations for the user:

- Generating recommendations based on keyword inquiries.
- Generating recommendations from basic questions with no context based on homework or quiz questions.
- Generating recommendations from multiple basic questions based on previous questions/material with which student may have struggled
- Generating recommendations from a student's email question regarding some concept from the course

The recommendations from the design tool are compared to what is typically expected as a result from the TA's/instructor or the students themselves. The key factors influencing the design tool's recommendations utilizing latent semantic analysis are the reduced dimensions matrix size and the amount of data available to search.

For data, the design tool will be analyzed using just the table of contents from various resources, as well as from full text and transcripts from available resources. There are many different dimensions to choose among for initial search results of LSA. The design tool uses 32 dimensions for the Table of Contents searching and 64 dimensions to start for the full text search. In general, the lower number of dimensions used, the more induction is present in LSA's recommendations; the higher number of dimensions used, the more keywords are directly influencing LSA's recommendations. The initial dimensions used by the design tool were chosen based on matching typical TA and the instructor recommendations for the Digital Design course from 2014 – 2016 for students who were experiencing trouble with the related query. The dimensions can be and are adjusted per user as needed based on user feedback as more course material is covered. As a design tool deployment option, the course instructor, TA's, or course developer can use a series of test-cases, similar to the ones presented in the following sections to test and set the initial dimensions as they see fit for the course offering if the default dimensions are not ideal. This is equivalent to performing unit-testing for major applications and allows users to fine-tune their product. The advantage here is that the code itself is not modified; only the lsa search dimensions are adjusted.

LSA Search Results and Performance

As mentioned previously, many of the source recommendations that the Automated Personal Advisor suggests are basic recommendations, which can typically be provided manually by an instructor or TA in the matter of a couple minutes or less in each case. It is important to keep the aggregate time involved over the entire course and for several hundred or more students to fully appreciate the amount of time saved. More in-depth recommendations are also possible based on the analysis of the data from each student's past performance, which is typically much harder to accomplish on an individual basis manually for all students in a single offering of a course due to time restrictions and other pedagogical priorities.

Basic Recommendations

On the minimal level, it is expected that the design tool can make recommendations to the user when they know what they are searching for based on some keywords or a homework question. This can be accomplished in many ways. LSA is shown to work satisfactorily for these cases below.

Utility of Searching Only Titles/Tables of Content

Latent Semantic Analysis can be performed on datasets as small as just titles without content. In this case, the chapter and sub-chapter headings are entered into our data structure. For this experiment, the individual chapter titles of two course books typically used for the Introduction to Digital Design: **Digital Design with RTL Design, VHDL, and Verilog** and **Digital Design and Computer Architecture: ARM Edition**. This provides 358 sources and 157 shared keywords. These number of sources and keywords will vary depending on course and how the instructor or course developer choose to separate sources and documents.

Basic Keyword Searching

As a trivial test case, the first search tested is representative of a student or TA browsing the table contents or index of the course books when they are looking for something specific and know the keyword:

Search query 1:

"Switches"

Top 3 results:

1. Ch 2.2 - Switches - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.993279296076
2. Ch 2.2 - The Amazing Shrinking Switch - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.993279296076
3. Ch 7.3 - Programmable Interconnects (Switch Matrices) - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.988068806503

The top 2 results can be reproduced by a student or TA looking at the index of the digital design book referenced. The 3rd result is not part of the index reference but is still a feasible recommendation depending on context.

Search query 2:

"Truth Tables"

Top 3 results:

1. Ch 2.6 - Truth Tables - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.980705006435
2. Ch 7.3 - Lookup Tables - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.950924244709
3. FSM Design - State Minimization using Implication Tables (Student Presentation) - Online Course Example Video
   Match Amount: 0.725737365133

The top result here overlaps with one of the sources from the digital design book's index as well. The remaining results are appropriate depending on context.

Search query 3:

"Digital Systems"

Top 3 results:

1. Ch 1.1 - Digital Systems in the World Around Us - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.986751937912
2. Ch 1.2 - The World of Digital Systems - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.986751937912
3. A. Digital System Implementation - from TOC of Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.985686947325

All three search results are appropriate and are presented as an anecdote that LSA can be used similar to a typical search engine with the correct keywords or searching through a book's index for some hint on additional study material that may not have been required or mentioned during lecture and discussion. It is important to note that indices of course books are typically created manually by the author; the ability to find overlapping references through a simple algorithm, especially one with no contextual knowledge of the material, shows promise that the algorithm can be used for more in depth recommendations. If required by the course developer, these results can be further fine-tuned through adjustment of the reduced dimensions matrix or including more data for the search algorithm to process. Alternatively, searches can be performed on specific resources for more customization as follows: for example, only the course books to which the student has access versus all resources available for the course or the top recommendation from each type of resource (top recommendations from each course book and top online videos).

Speed Comparison of Keyword Searching Manually vs through the Design Tool

In this novel case, manually looking through the index of a course book takes anywhere from several seconds to a minute. The design tool can perform the same search nearly instantly (on average .03 seconds for the above queries + time it takes to enter the query).

Homework Help

A student may have a question on their homework assignment and not sure which resource is best for them. In this case, if the student knows what to search for, they can input the appropriate keywords and get similar results to the trivial case. Alternatively, they can attempt to enter the entire question in a search engine or check through the table of contents manually to decide which chapter section may be the most relevant for their needs.

This is a typical situation where students may not know what references they should check. An email may be sent out to the TA, with a response received later that day or the next day. Having an algorithm that can perform a simple suggestion based a sentence or question that has potentially many keywords would be useful for the student & can help them better manage their time. A simple keyword search algorithm may not be satisfactory, even for a simple situation like this, since there are many keywords.

LSA is based on co-occurrence of words and the relationship among them. Even with just chapter headings from a book, a practical suggestion can be made by LSA.

Search query 1 (a question from a Number Conversion homework):

"Numbers: Complete the following table by converting the following numbers: Convert Two's Complement Binary 01110101 to Decimal"

Top 3 results:

1. App B.2 - Real Number Representation - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.919458152824
2. Ch 4.6 - Subtractors and Signed Numbers - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.917303568825
3. Ch 4.6 - Subtractor for Positive Numbers Only - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.917303568825

Search query 2 (a question from an Arithmetic homework):

"Arithmetic: Subtract the following 7-bit two's complement numbers"

Top 3 results:

1. Arithmetic - Two’s Complement Addition, Two’s Complement Subtraction - Online Course Example Video
   Match Amount: 0.878276991056
2. Ch 4.6 - Representing Negative Numbers: Two's Complement Representation - from TOC of Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.867276055158
3. Arithmetic - Two’s Complement Multiplication (Student Presentation) - Online Course Example Video
   Match Amount: 0.723435274477

Search query 3 (a question from a Logic Gates homework):

“Logic Gates: Create the error detection circuit for the two's complement adder / subtractor shown below.”

Top 3 results:

1. 1.5 Logic Gates - from TOC of Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.790227170643
2. 2.4 From Logic to Gates - from TOC of Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.790227170643
3. Logic Gates - Online Course Concept Video
   Match Amount: 0.790227170643

These results are typical of a reference a TA may provide through email support and can be provided automatically by the design tool. The student saves the time of needing to wait for an answer from the TA or instructor, and the instructor/TA saves time of needing to check the book for a relevant reference for the student (or giving a generic recommendation of “read chapter 4 from the book”). Some typical homework feedback for this homework from graders include “Good job, but your optimized circuit has more gates (than necessary)” and “you should have used a different equation”. These tips are helpful, but they do not refer the student to a specific subchapter or section. The automated education system can provide these references automatically for the grader to include to supplement the comments. Alternatively, the grader comments can be bypassed altogether and the automated education system can be responsible for providing feedback, which would save time for the grader and allow them to spend more time for in-person interactions with students.

Speed Comparison of Homework Help from an Instructor/TA vs through the Design Tool

From the student perspective, the time a student typically waits for an answer from a professor ranges from hours to days. Receiving a response within a 24 hour period is considered reasonable and can be accomplished under most cases. During this time, a student is typically stuck on the concept and can't move on until a hint or suggestion is provided. The design tool's ability to provide an answer almost immediately (search results for these queries took between .03 seconds on average) saves the students time.

From the instructor and TA perspective, emails can take anywhere from 30 seconds to several minutes to answer, even for just simple course book or video reference responses as shown in the previous sections. A design tool that can automatically suggest the proper references saves this time for the instructor and TA's per student. In a class of 200 students, with each student potentially sending several emails to the instructor or TA's throughout a 10 week course, the amount of time saved could range easily in terms of hours. These hours can now be dedicated to discussing more complicated concepts directly with students.

Quality Comparison of Homework Help from an Instructor/TA vs through the Design Tool

In relation to quality, the recommendations provided by the design tool were initially normalized through choosing the proper dimensions to be equivalent to recommendations left for students by the graders on their homework. Typical responses from graders mentioned the concept the students should review, which left students to search for sources on their own similar to the keyword searching section. Other typical responses may include a chapter or section reference for common problems that many students may have missed, similar to the homework help section results. Based on student feedback in lecture, discussion, and surveys ([8] and Appendix), this feedback is helpful for them.

Advantages and Limitations of Searching only through the Table of Contents

LSA can make simple suggestions based on the table of contents of books and other sources. Searches are also quick since there is not a very large search space. However, for more in depth recommendations, more information is required. Processing longer queries is more effective if there is more content to search through. In the next section, an analysis of more search queries through a significantly larger data set is analyzed.

With the limited number of words for each document since the design tool only uses titles for these searches, some results are very close in terms of matches, such as some of the results from queries 1 & 3 in the previous section. If more content is provided in the data structure, these matches would differentiate more significantly, as we will show later through full text searching.

More In-Depth Recommendations

In order to provide more in depth recommendations, more content must be provided. In the following experiments and results, the design tool searches through full text documents, which are split by chapter sections. Videos are split into approximately 5 minute sections to mimic small chapter section content.

Utility of Searching Full Text with LSA

Search result quality and detail can be increased by providing more data for the data structure. Utilizing LSA as a foundation allows the design tool to provide comparable suggestions as TA's or instructors in response to student performance on online quizzes, homework, and email inquiries. The main advantage of using LSA vs basic keyword searching or traditional artificial intelligence algorithms is the ability to implement and adapt the search space without requiring application-specific/expert knowledge of the search domain space.

As mentioned previously, LSA is implemented based on mathematical observations of words' relationships among each other but not of the meanings of the words themselves [38]. It is important to note that no contextual knowledge is needed for implementing LSA for this search space; however, it wouldn't hurt to add some knowledge where a developer may be able to make efficiency improvements. This concept is very powerful and allows the design tool to be deployed and implemented across multiple disciplines and applications without requiring extensive reworking of the core algorithms used to make recommendations. For the following searches, the design tool is accessing over 440 recommendation sources based on chapter sections and videos with over 3200 shared keywords among the documents.

Basic Keyword Searching

Revisiting the trivial keyword search case, we now look at the results of searching through full text versus just the table of contents.

Search query 1:

"Switches"

Top 3 results:

1. Back to the Future - Digital Design 101 Online Lecture Concept Video
   Match Amount: 0.658647395621
2. Ch 2.2 - Switches - The Amazing Shrinking Switch - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.595895368071
3. Ch 1.2.1 - From Zero to One - The Art of Managing Complexity - Abstraction - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0. 58407336713

The search result of the "Back to the Future" video did not show up in the basic table of contents only search since there wasn't enough context for the LSA algorithm to determine a match. However, including the transcript of the video puts the centroid vector of the search query closer to the lecture video than the other results because of the use of the word "switches" in the actual content/body of the reference. These results can be customized and adjusted by changing the dimensions used in the reduced dimensions matrix to provide more personalized results for a user. This will be analyzed in a later section.

Search query 2:

"Truth Tables"

Top 3 results:

1. Ch 2.6 - REPRESENTATIONS OF BOOLEAN FUNCTIONS - Standard Representation and Canonical Form - Standard Representation - Truth Tables - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.801330374462
2. Ch 3.4 Finite State Machines - Sequential Logic Design - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.781265198014

3. Ch 2.6 - REPRESENTATIONS OF BOOLEAN FUNCTIONS - Truth Tables - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.778448162799

Note in these results, a source covering finite state machines is a closer match than one specifically for truth tables. Again, this is acceptable and expected because the design tool is using a reduced dimension matrix of 64 (from over 440), which performs more induction/guessing than a straight keyword search. The lower the reduced dimension matrix, the more induction is performed.

Search query 3:

"Digital Systems"

Top 3 results:

1. Ch 1.2 - Digital versus Analog - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.821161419019
2. Ch 1.2 - Digital Circuits are the Basis for Computers - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.81834311729
3. Ch 1.3 - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.801406738837

The results here are different than when searching through just the Table of Contents of the course books. This makes sense since there are many more keywords and sources to search through now (446 sources with 3270 keywords for this search). The results, although different, are still relevant for a generic search for information on "switches". As stated previously, these results can be further customized depending on user preference (for example, which resources the user found most helpful from their search) and adjusted for future results.

Homework Help

Now with a larger data set to search through, the sample search from the Homework Help section previously is shown again for a larger search space using the same LSA algorithm with some adjustments for the dimensions matrix given the larger number of sources and keywords.

Search query 1 (a question from a Number Conversion homework):

"Numbers: Complete the following table by converting the following numbers: Convert Two's Complement Binary 01110101 to Decimal"

Top 3 results:

1. Ch 1.4.6 - From Zero to One - Number Systems - Signed Binary Numbers - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.855062505572
2. Ch 1.4.6 - From Zero to One - Number Systems - Binary Addition - Example 1.12 - Adding Two's Complement Numbers - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.854625402292
3. Ch 1.4.6 - From Zero to One - Number Systems - Binary Addition - Example 1.10 - Two's Complement Representation of a Negative Number - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.852118948145

Search query 2 (a question from an Arithmetic homework):

"Arithmetic: Subtract the following 7-bit two's complement numbers"

Top 3 results:

1. Ch 4.6 - Datapath Components - SUBTRACTORS AND SIGNED NUMBERS - Representing Negative Numbers: Two's Complement Representation - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.92005224396
2. Ch 1.4.6 - From Zero to One - Number Systems - Binary Addition - Example 1.13 - Subtracting Two's Complement Numbers - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.900746586703

3. Ch 1.4.6 - From Zero to One - Number Systems - Binary Addition - Example 1.12 - Adding Two's Complement Numbers - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.896877068848

Search query 3 (a question from a Logic Gates homework):

"Logic Gates: Create the error detection circuit for the two's complement adder / subtractor shown below."

Top 3 results:

1. Ch 6.4 - Optimizations and Tradeoffs - DATAPATH COMPONENT TRADEOFFS - Faster Adders - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.813189693034
2. Ch 4.3 - Datapath Components - ADDERS - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.768308701337
3. Ch 4.3 - Datapath Components - ADDERS - Adder - Carry-Ripple Style - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.737199497009

The results returned now are more specific than the basic search from the previous section using the same query; this is sensible since there is much more data to search through. The individual utility of the results of the search depend on context; in particular, the user performance level can be taken into account, and the search dimension matrix can be adjusted depending on user preference. This will be analyzed and reviewed in a later section.

Training/adapting LSA for better individualized results

Since students do not all study or learn in the same manner, results for students who struggle with the same types of questions (for example the Numbers and Logic Gates query in the previous section) may vary in utility. If the average student in this case finds that the results are acceptable, no further customization is needed. However, if a student finds that result #2, which covers Adders, is more useful for them than result #1, there are different ways to address future search results. By collecting information on what a student may feel works better for them (perhaps through a star rating system, similar to contemporary social media review systems), more back-end adjustments can be made.

Creating an individualized experience based on user preference and feedback

For the search case in the previous section, the design tool started at a reduced dimension space of 64. If the student prefers result #2 vs result #1 (and didn't find #1 useful at all), they can rate result #2 with 5 stars out of 5 & result #1 with 1 star. The design tool can then perform additional metric analysis and searches through different dimension spaces in order to adjust the match quality compared to the original search. Once a new dimension space is found that correlates closer to the user preferences so far, that new dimension space can be used.

One possible implementation can be incrementally adjusting the search space until result #2 has a higher match quality than result #1, then confirm with the user with the new search results since other sources may rank higher than both previous results now. This can be done directly by the user through a specific "settings" menu option or done automatically through future searches.

The design tool uses a binary search to determine the next dimensions: 32 (halfway between 0 and 64 dimensions) and 255 (halfway between 64 and 446 total dimensions). With 32 dimensions, the "subtraction" result was still number 1, so the design tool looks toward 255 dimensions. With 255 dimensions, the "subtraction" result match quality lowered, but so did the “adder” result. However, there are multiple references to different sections of Ch 4.3 here & a more specific section is rated higher than the “subtraction” result:

1. Ch 4.3 - Datapath Components - ADDERS - Adder - Carry-Ripple Style - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.71205977454
2. Ch 4.6 - Datapath Components - SUBTRACTORS AND SIGNED NUMBERS - Representing Negative Numbers: Two's Complement Representation - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.700976345769
3. Ch 5.2.2 Subtraction - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.674633826654
   …
7. Ch 4.3 - Datapath Components - ADDERS - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.633823054669

Ch 4.3 is split so that the design tool can suggest more specific sections and sub-sections than a more general response. Before presenting results to the user, the design tool can check further dimensions for fine-tuning.

The next 2 dimensions based on a binary search is 159 and 350. 159 dimensions increases the distance between the “subtraction” and “adders” result while 350 puts “adders” ahead of “subtraction” in the results. The results for 350 dimensions are as follows:

1. Ch 4.3 - Datapath Components - ADDERS - Adder - Carry-Ripple Style - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.711949464959
2. Ch 4.6 - Datapath Components - SUBTRACTORS AND SIGNED NUMBERS - Representing Negative Numbers: Two's Complement Representation - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.700760190701
   …
5. Ch 4.3 - Datapath Components - ADDERS - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.633050856325

6. Ch 5.2.2 Subtraction - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.631385081357

For this particular user, the design tool may now start using 350 dimensions for the search space and fine-tune as required based on user feedback. Conventionally speaking, higher number of dimensions results in more direct keyword searches whereas lower number of dimensions make more inductive guesses [38]. The number of dimensions a user prefers may change throughout the course and can be monitored through user feedback. Additional metrics can be used to create a custom user experience, which is discussed in

Chapter 6: Future Work.

Homework Help with More Context and Individualized Dimensions

Now let's take it a step further and add some context into our query. First, we take a look at a student, Student A, struggling with an RTL component problem (Logic Gate search query from the previous section). Now suppose this is the same student who wanted help from the previous search needs help later on when working on a Adder/Subtractor (also likely to be implemented using two's complement numbers). With this new context, we have an updated query (combining both questions) and updated results. Suppose that the student preferred the results of the LSA algorithm so far (64 dimensions).

Search query 1 (combined questions for Student A with 64 dimension search matrix):

"Numbers: Complete the following table by converting the following numbers: Convert Two's Complement Binary 01110101 to Decimal. Logic Gates: Create the error detection circuit for the two's complement adder / subtractor shown below."

Top 3 results:

1. Ch 5.2.2 Subtraction - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.795448961288
2. Ch 4.3 - Datapath Components - ADDERS - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.746620032387
3. Ch 4.6 - Datapath Components - SUBTRACTORS AND SIGNED NUMBERS - Detecting Overflow - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.742917197201

With the new context, the query now has an updated centroid vector, which aligns closer to Ch 5.2.2, which reviews two's complement subtraction rather than providing a result going over optimizing an adder to be faster in Ch 6.4 for the Digital Design book, which was a valid result without any additional context. The induction done by LSA here is very promising considering no engineering-specific contextual knowledge was added to create these results. Further optimization can be fine-tuned by both students and instructors and will be discussed in the following section.

In another case, we may have another student, Student B, who may have a similar issue with the homework, but preferred different results, such as those outlined in the previous section, resulting in a 350 dimension reduced matrix. This student, using the same query, gets different results:

Search query 2 (combined questions for Student B with 350 dimension search matrix):

"Numbers: Complete the following table by converting the following numbers: Convert Two's Complement Binary 01110101 to Decimal. Logic Gates: Create the error detection circuit for the two's complement adder / subtractor shown below."

Top 3 results:

1. Ch 4.3 - Datapath Components - ADDERS - Adder - Carry-Ripple Style - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.711949464959
2. Ch 4.6 - Datapath Components - SUBTRACTORS AND SIGNED NUMBERS - Representing Negative Numbers: Two's Complement Representation - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.700760190701
3. Ch 6.4 - Optimizations and Tradeoffs - DATAPATH COMPONENT TRADEOFFS - Faster Adders - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.67164185721

When adding in context and expecting certain results, user feedback becomes very important in order to adjust the LSA dimensions appropriately per user. Searching with the same queries through the data structure with different dimensions results in different results based on user feedback. As the user increases their participation and feedback, the results presented by the design tool for that user should adapt accordingly.

For this experiment, these results are presented to show that it is possible to create more meaningful results with relatively low overhead that may save both student and instructor time in finding answers to their questions.

Speed Comparison of Homework Help with Context from an Instructor/TA vs the Design Tool

This detail of homework help presented in the previous section is not typically conducted pre-emptively in a classroom environment with hundreds of students simply due to time restrictions and feasibility. This process can potentially take an instructor or TA several minutes per student per query to arrive at a suggestion or recommendation once past homeworks are brought into context.

With the help of the design tool, this information can be presented to students pre-emptively, or to the instructor/TA before meeting with a student to maximize the small amount of time available for in-person meetings. The algorithm itself only takes approximately .03-.1 seconds to process queries similar to those presented in the previous section.

Quality Comparison of Homework Help with Context from an Instructor/TA vs Design Tool

In relation to quality, the recommendations provided by the design tool here were initially normalized through choosing the proper dimensions to be equivalent to recommendations left for students by the graders on their homework, similar to the homework help section. The power of this tool comes in freeing up time that was previously spent manually searching through a student's homework history to best determine with which sections the student needs help. The available time can now be better utilized to help students with more difficult concepts.

Enhancing interaction between Instructor/TA and Student

In addition to pre-emptive automated recommendations, the design tool can also serve as a virtual assistant for the instructor or TA when responding to student inquiries. Although finding relevant references for students doesn't take typically take a significant amount of time for each instance, as class room sizes grow, every minute saved adds up and matters. The design tool can be utilized to process student emails and message board posts, just like they were shown to process homework questions in previous sections. The results can be provided to the student automatically, or included in the electronic communication from the instructor or TA.

Enhancing Email and Forum Responses with Automated Recommendations

The design tool can also be used to process student inquiries from email and provide potential resources for the student to view. These next queries are examples of student email questions:

Student query 1 (email inquiry):

"I am still having optimizing the full adder into NAND, NOR, and NOT gates from homework 10. Do you have time after class on Monday to work out the problem with me? Unfortunately, I am unable to make your office hours due to class."

Top 3 results:

1. Ch 6.4 - Optimizations and Tradeoffs - DATAPATH COMPONENT TRADEOFFS - Faster Adders - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.808190732824

2. Ch 4.3 - Datapath Components - ADDERS - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.677069919044
3. Technology Mapping (5m - 10m) - Digital Design 101 Online Lecture Concept Video
   Match Amount: 0.675524846715

For this situation, a manual reply is required. However, having reference results available for the instructor, which can be done automatically by the design tool, will save the instructor time from needing to look up appropriate references for the student to review before coming to an in-person appointment. For this particular case, any of the 3 results would work for references since they all cover converting gates. These can be attached automatically at the end of the email, or copy & pasted by the responder manually.

Student query 2 (posted on course message board):

"Can someone explain and post the timing diagram of Register file timing question of HW12?"

Top 3 results:

1. Ch 4.10 - Datapath Components - REGISTER FILES - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.921716867257
2. Ch 5.5.5 Register Files - Digital Design and Computer Architecture ARM Edition
   Match Amount: 0.787264730051
3. Ch 3.2 - STORING ONE BIT-FLIP-FLOPS - Basic Register - Storing Multiple Bits - Digital Design with RTL Design, VHDL, and Verilog
   Match Amount: 0.782060948493

Again, a manual reply was required here, but including the relevant course book references manually took some time, which could have been utilized to answer more emails. The design tool provides a quick and relevant reference that can be used by the instructor/TA, or even provided automatically when the user chooses to post a question on the message board. If the references answered the student's question, the question could be skipped completely, or posted with reference as a solution to allow other students who had similar questions to review the material without additional intervention.

Chapter 6: Future Work

The Active Learning Personal Advisor has been shown to be promising at providing recommendations for both students and instructors, allowing for more efficient use of time. The next logical steps for the design tool are through further automation and customization. The most ubiquitous method for improving the design tool is through integration with the University Course Portal to provide a fully integrated and fluid experience for both students and instructors.

Integrating into a University Course Portal

Students at the University of California, Irvine access their grades and many course materials through the Electronic Educational Environment (EEE) [47]. Currently, the design tool receives grades and data from the message board through manual input. Obviously, integrating the design tool into EEE would allow for more automated processing of grades and recommendations, directly through the University Portal vs through a separate web site. Instructors and TA's would only have to enter grades into one site/page and can receive student reports directly in the course portal.

The digital and web-based nature of the Active Learning Personal Advisor naturally fits into any typical university course portal. I plan to work in conjunction with the Center for Engaged Instruction [48] as well as the Office of Information Technology [49] at UC Irvine to implement expanded automated education system prototypes for more engineering courses. As more courses are added to the automated system, more relevant and useful results and reports can be generated through searching through larger datasets.

Expanding Automated Grading

Student performance profiling is currently automated through the self-check quizzes consisting of multiple-choice/fill-in-the-blank automated question grading. More accurate profiling is conducted once graders, TA's, and instructors submit grades for homework and exams, which typically have questions with wider ranges of difficulty. Expanding the variety of questions available in the self-check quizzes that allow for more complicated questions allows for a quicker user feedback and recommendation loop. For the digital design courses, automatically graded questions currently consist of fill-in-the-blank, multiple choice, and true-or-false questions. A larger variety of questions can be included through allowing circuit connection completion and circuit design related interfaces. A graphical interface allowing the user to add, remove, and edit components would be required. This type of interface can be used as a foundation for other engineering courses as well.

For the Digital Logic Design course, creating a tool to allow for circuit building and verification would allow for automated grading of circuits. This will drastically cut down on grading time for TA's and instructors, as well as allow students to have instantaneous feedback on their designs. These more advanced tools will allow for higher levels of critical thinking from students while maintaining ease of grading, which will maintain scalability as enrollment inevitably increases in the major and school as a whole.

Early Intervention through Linear Progression Modeling

Some pre-emptive intervention is possible through the Active Learning Personal Advisor. When students answer a series of questions wrong, recommendations can be provided quickly. Also, when a student has been performing below expectations, the design tool can be set to notify the instructor automatically. However, potential signs that a student will perform worse without early intervention can be potentially predicted through linear progression modeling. If successful, the design tool would be able to recognize patterns based on available study habits and patterns of students through the integrated university portal based on a variety of metrics available in the design tool data structure. Linear progression modeling can compile data on when students (current and past) access their homework, when they turn it in, when they view online slides, videos, how often they take self-check quizzes, how soon they access referenced material after a recommendation email is sent, etc., and predict if the student needs to change their habits or intervention may be required. This type of modeling becomes even more effective in conjunction with the concept of cross-course searching and long-term profiling, which are highlighted in the next section.

Integrating an early intervention model can greatly aid both instructors and academic counselors. The automated education system can perform routine quarterly, weekly, or even nightly checks of student progress. Once a red flag situation occurs with a student profile, an email can be dispatched to the proper educator or counselor with appropriate information from the data structure to give the educator a quick snapshot overview of the affected student. This information will help streamline any necessary in-person discussion of the academic progress of current students, improving student interaction and retention with minimal overhead increases.

Improving Overall Search Results through Crowd-Sourcing Techniques

Currently, the user feedback system is primarily aimed toward improving the experience for that particular user. However, there is continuing research with respected to the idea of gaining wisdom from crowds [50], or for this tool's case, better search results. The idea is that if a large crowd is asked to answer a question, the average or most popular answer has a high probability of being correct; this has been shown to work for very well for easy multiple-choice questions [51]. By utilizing this technique through aggregating results from user feedback, the overall weighting of keywords and documents can be adjusted for entire courses, resulting in better search results for more students.

Cross-Course Searching and Long-Term Profiling

Once integrated into the University Portal, it would be possible to allow the design tool to search through pre-requisite or related courses the student has taken. This will allow for much deeper recommendations for students and performance profiling can be conducted across a student's entire academic career. Recommendations from the design tool with the ability to access assignments and references available to the student from pre-requisite courses will be especially helpful for students who may need to review those materials. Cross-course referencing can be added through referring to the course catalogs for universities or directly from the instructors and course developers.

With long-term profiling, other tools, such as linear progression modeling are much more effective and can be used to self-adjust since the design tool can compare actual student progression through several courses to its predictions of outcomes. A larger histogram is available for the design tool to utilize to analyze student performance, giving the design tool a more comprehensive view of which recommendations and results were most effective.

With long-term profiling, LSA can potentially even be used to make recommendations for elective courses in which the student may be interested. Academic counselors and course planners can utilize the aggregated data for student placement in future courses based on prospective recommendations.

Increasing Context Awareness of LSA

LSA, by default, has no contextual knowledge of the data it searches through. For this application of LSA, it has not created any problems since the dataset is controlled. However, with plans to expand the use of LSA, adding context awareness would be prudent. One such way is to convert from a keyword/non-grammatical data structure to an n-gram model, which stores words in sequences. Additionally, the issues of synonyms and polysemy recognition can be addressed by explicitly training the design tool to recognize synonyms and interpreting specific words a certain way.

A novel example of adding synonyms for a digital design course would be to include words like selector, multiplexor, and mux as synonyms, since they are. These words are often used interchangeably in books and online content. The existing automated education system treats these words as separate words, which does not affect most basic searches; however, future searches, especially once more homework and self-check quiz questions are added, may be affected by a lack of context awareness through using LSA.

Including Internet References into the Data Structure Automatically

Currently, the data provided to the data structure is entered manually and provided directly by the instructors or course developers. However, depending on the course, there may be additional useful references. Arbitrarily and randomly adding sources from Internet searches, although possible, may not be ideal. Instead, the instructor, course developer, or in some cases the students themselves, may add a site name, journal, or discussion thread they find relevant to the course. With the web site URL, the design tool can parse the digital data automatically and expand its data structure incrementally.

For example, for the digital design courses offered, many students have found that certain YouTube channels had helpful supplementary videos that aided in their understanding of the class material. These videos can be transcribed and indexed by the automated education system, which will include them in searches and results for future offerings. Adding these sources would greatly increase the amount of data available for searching, which will also help students find additional and relevant resources in their research.

Chapter 7: Summary of Contributions

The Active Learning Personal Advisor contributions to large-scale engineering education can be summarized as follows:

1. Categorize and profile students based on performance level
    a. Allows opportunity for future recommended studies and assignments for students automatically as content becomes available
    b. Allows instructors and TAs to quickly assess individual student performance prior to or at the beginning of in-person meetings
    c. Aggregate automatic student performance profiles help instructors identify overall class strengths and weaknesses and help TAs identify overall discussion section strengths and weaknesses, helping guide topics covered in lecture and discussion
2. Automatically provide timely recommendations to students to facilitate self-motivation
    a. Enhanced self-check quiz feedback
    b. More detailed homework and topic review comments
    c. Aid instructors in creating more detailed replies in response to student queries via email and other online communication
3. Adapt search results over time for individual students through a user feedback system
    a. Different students will receive different results based on their own learning preferences
    b. Recommendations are student-centered and individualized through customization

Automated Categorizing and Performance Profiling

The Active Learning Personal Advisor categorizes and profiles students by their performance levels with techniques in mind from other disciplines to maximize not just performance but also motivation and self-satisfaction; existing tools for managing MOOC's follow traditional methodologies and provide the same experience for everyone, hindering motivation, self-satisfaction, and inevitably performance for many students.

Initial assessments can be calculated through available self-check quizzes, which are written in a way to allow for simple automated grading. Feedback and recommendations can be supplied to users nearly instantaneously, maintaining and encouraging active learning. Further assessments can be included as graders, TA's, and instructors upload assignment and exam grades, allowing the design tool to continue to build the performance profile of student. Once the automated education system is integrated into a larger university course portal, the overhead for entering manual data into the automated system is removed, saving even more time in long term implementations of the Active Learning Personal Advisor.

These automatically generated profiles are more useful than current course management systems, which only offer a very broad overview of student performance. The automated education system can provide detailed suggestions and keywords with which each student is struggling. From there, an instructor can quickly assess the student's needs and make further recommendations at in-person meetings.

Just as individuals in any given course have differing levels of aptitude and performance, students as a group vary from course offering to offering. Aggregate class performance profiles can also be generated through the automated education system. Similar to the individual reports, keywords of strengths and weaknesses for an entire lecture or discussion can be provided. As an example from the digital design courses covered in this dissertation, many students from a previous offering were struggling with two's complement addition, as indicated from homework graded by the TAs for the course. However, since the homework consisted of several different topics and were graded by multiple graders and TAs, this information was not immediately communicated to the instructor by the graders or TAs. When this information was entered into the automated education system, a performance profile is created. This profile includes information containing the most missed questions, keywords of topics related to those questions, and references for further studying for students can be passed on from the instructor. More information can be included by the course developer. This will help the instructor and TA determine what concepts and problems need to be addressed and in what order.

Timely Recommendations

Students prefer detailed help when possible. However, with increasing enrollment, details, such as recommending specific chapter subsections, are not always practical to provide. At best, a TA may be able to recommend a subchapter for all homework questions during a particular week. An advantage of an automated education system here is that it can recommend a specific subsection related to a specific self-check quiz or homework question almost instantaneously, as seen in Chapter 5: Experiments and Results. Recommendations can be made based on self-check questions and any other graded material automatically.

Customizing Search Results over Time for Individual Students

Because students have varying skill levels, they may need different recommendations, even when missing similar questions. The automated education system allows users to provide feedback on search results, similar to popular online search engines and social media sites adjusting targeted advertising for users. The Active Learning Personal Advisor adjusts the amount of induction it makes during recommendations by changing the dimensions of searching through LSA. A binary search is conducted to performance basic rearrangement of results. As the user interacts with the automated education systems, search results will become more unique to the user.

Furthermore, once the automated education system is integrated into a university course portal, long-term customization is possible. For example, if a student performs exceptionally well during the introductory digital design course, their search results and recommendations will be different than students who were struggling with the same course in future offerings since the student performance assessment information is carried on from one course to others in which they are prerequisites. Since there will be multiple courses and much more feedback per student over the course of their academic career, search results can be more individualized per user. Additional customization through keyword and document weighting is possible as more data is available to adjust in the data structure.

This information can also be useful for instructors at the beginning of the quarter. The automated education system can provide a generalized aggregate report of overall student performance assessments, highlighting strengths and weaknesses based on their performance in prerequisite classes. As an example, assume that students are enrolled in a digital design lab course that is a follow up of the prerequisite digital design course taught by another instructor. In a typical situation, the lab course instructor must spend some time manually assessing student strengths and weaknesses.

However, with the automated education system, if most students did not perform well on creating larger selectors from smaller selectors in the prerequisite digital design course, this information including keywords and references to homework, self-check quizzes, or exam questions can be included in the initial aggregate report to the instructor. This will indicate to the instructor that they may want to spend extra time reviewing building larger selectors from smaller ones. Conversely, if most students performed exceptionally well in a certain topic, like computing the minimal sum-of-products through k-maps indicated by keywords and references from the automated education system, the instructor would know that they can spend less time on this topic in the follow-up course.

Summary

Automating and providing support for the lower-order needs of students frees up precious time, typically adding up to potential savings of hours for students allows both students and instructors to utilize the limited amount of in-person time they have available to be more productive. Student performance profiles and recommendations are also shared with the TA's and instructors in form of reports easily accessible from the design tool data structure, allowing instructors to more quickly pin-point student issues and needs. The recommendations can be fine-tuned by both students and instructors through user feedback methods rating the recommendations and reports.

The Active Learning Personal Advisor accommodates larger scale instruction while minimizing overhead and time commitment from both students and instructors. The data structure provides a meaningful dataset for the design tool, which uses the data in algorithms that generate relevant reports containing suggestions and reviews of the student's work based on each individual's progress. This automated system is integral in making a hybrid methodology for large scale courses with minimal instructor intervention for lower level needs possible while focusing on satisfying the metrics proposed. As contemporary education moves forward with larger enrollments and tighter budgets, automated learning design tools will move from being novel luxuries to integral learning models for success.

Appendix

Study Habits/Resources used Survey for students in a Digital Design course (Fall 2015 – 110 of 123 students responded):

Please rate how each of the following materials has been a part of how well they helped you learn about the current course material. There are no right or wrong answers on this one, we just want to see what works for which students (disagree = not helpful, agree = helpful, no opinion = didn't use this material).

1. Lectures

2/110	2%	No Opinion
1/110	1%	Strongly Disagree
1/110	1%	Disagree
63/110	57%	Agree
43/110	39%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

2. Discussions

14/110	13%	No Opinion
0/110	0%	Strongly Disagree

2/110	2%	Disagree
42/110	38%	Agree
52/110	47%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

3. Homework

3/110	3%	No Opinion
1/110	1%	Strongly Disagree
3/110	3%	Disagree
53/110	48%	Agree
50/110	45%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

4. Lecture Videos

6/110	5%	No Opinion
3/110	3%	Strongly Disagree
13/110	12%	Disagree

58/110	53%	Agree
30/110	27%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

5. Sample Problem Videos

12/110	11%	No Opinion
2/110	2%	Strongly Disagree
11/110	10%	Disagree
55/110	50%	Agree
30/110	27%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

6. Self-check quizzes

23/110	21%	No Opinion
5/110	5%	Strongly Disagree
10/110	9%	Disagree
54/110	49%	Agree

18/110	16%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

7. Coursebook chapters

65/110	59%	No Opinion
2/110	2%	Strongly Disagree
14/110	13%	Disagree
24/110	22%	Agree
5/110	5%	Strongly Agree
110/110	100%	# of responses to this question
0/110	0%	No answer selected

8. Coursebook questions

73/110	66%	No Opinion
4/110	4%	Strongly Disagree
9/110	8%	Disagree
19/110	17%	Agree
5/110	5%	Strongly Agree

110/110	100%	# of responses to this question
0/110	0%	No answer selected

Online Communications Responsiveness/Helpfulness Survey for students in a Digital Design Course (Fall 2015 – 111 of 123 students responded):

Please rate how each of the following materials has been a part of how well they helped you learn about the current course material. There are no right or wrong answers on this one, we just want to see what works for which students (disagree = not helpful, agree = helpful, no opinion = didn't use this material/resource).

1. Email communication w/ the TA or instructor via email regarding course material in terms of response time.

46/111	41%	No Opinion
0/111	0%	Strongly Disagree
0/111	0%	Disagree
28/111	25%	Agree
37/111	33%	Strongly Agree
111/111	100%	# of responses to this question
0/111	0%	No answer selected

2. Email communication w/ the TA or instructor via email regarding course material in terms of helpfulness/guidance.

50/111	45%	No Opinion
0/111	0%	Strongly Disagree
1/111	1%	Disagree
31/111	28%	Agree
29/111	26%	Strongly Agree
111/111	100%	# of responses to this question
0/111	0%	No answer selected

3. Homework comments in terms of helpfulness/guidance.

6/110	5%	No Opinion
4/110	4%	Strongly Disagree
18/110	16%	Disagree
58/110	53%	Agree
24/110	22%	Strongly Agree
110/111	99%	# of responses to this question
1/111	1%	No answer selected

4. Message Board interaction in terms of response time.

32/110	29%	No Opinion
0/110	0%	Strongly Disagree
1/110	1%	Disagree

41/110	37%	Agree
36/110	33%	Strongly Agree
110/111	99%	# of responses to this question
1/111	1%	No answer selected

5. Message Board interaction in terms of helpfulness/guidance.

24/111	22%	No Opinion
0/111	0%	Strongly Disagree
2/111	2%	Disagree
46/111	41%	Agree
39/111	35%	Strongly Agree
111/111	100%	# of responses to this question
0/111	0%	No answer selected

References

[1] U. o. C. O. o. t. President, "UC California feshman admissions for Fall 2003 - University of California Office of the President," University of California Office of the President, 6 November 2004. [Online]. Available: http://www.ucop.edu/news/factsheets/2003/admissions_summary.pdf. [Accessed 26 August 2013].

[2] U. o. C. O. o. t. President, "UC California freshman admissions for Fall 2013 - University of California Office of the President," University of California Office of the President, 17 April 2013. [Online]. Available: http://www.ucop.edu/news/factsheets/2013/fall_2013_admissions_table2.pdf. [Accessed 26 August 2013].

[3] U. o. C. A. S. -. T. S. Source, "Report Details Trends for Faculty Workload," University of California Academic Senate - The Senate Source, June 2013. [Online]. Available: http://senate.universityofcalifornia.edu/news/source/facultyworkload.june2013.html. [Accessed 26 August 2013].

[4] Coursera, "About | Coursera," 2016. [Online]. Available: https://about.coursera.org/. [Accessed 1 December 2016].

[5] Udacity, "About Udacity," 2016. [Online]. Available: https://www.udacity.com/us. [Accessed 1 December 2016].

[6] N. S. o. S. E. (NSSE), "2016 Topical Module: Learning with Technology," NSSE, Center for Postsecondary Research, Indiana University School of Education, Bloomington, IN, 2016.

[7] W. Oremus, "University Suspends Online Classes After More Than Half the Students Fail," 17 July 2013. [Online]. Available: http://www.slate.com/blogs/future_tense/2013/07/19/san_jose_state_suspends_udacity_online_classes_after_students_fail_final.html. [Accessed 12 August 2013].

[8] D. D. Gajski, Q.-V. Dang and W. He, "An Online Methodology for Individualized Education," in *2013 International Conference on e-Learning, e-Business, EIS, & e-Government (EEE'13)*, Las Vegas, NV, July 22-25, 2013.

[9] O. Delialioglu and Z. Yildirim, "Design and development of a technology enhanced hybrid instruction based on MOLTA model: Its effectiveness n comparison to traditional instruction.," *Computers & Education,* vol. 51, no. June 2007, pp. 474-483, 2008.

[10] J. Baker, "The 'Classroom Flip': Using Web Course Management Tools to Become the Guide on the Side.," in *11th Conference on College Teaching and Learning*, Jacksonville, FL, 2000.

[11] D. Ariely and K. Wertenbroch, "Procrastination, Deadlines, and Performance: Self-Control by Precommitment," *PSYCHOLOGICAL SCIENCE,* vol. 13, no. 3, pp. 219-224, 2002.

[12] Q.-V. Dang and D. Gajski, "Creating an Automated Learning Management Tool in an Engineering Course Based on Interdisciplinary Metrics," in *2014 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS 2014)*, Las Vegas, NV, 2014.

[13] Q.-V. Dang and D. Gajski, "Bringing In-Class Online: A Hybrid Solution," in *4th Interdisciplinary Engineering Education Conference (IEDEC 2014)*, Santa Clara, CA, 2014.

[14] G. V. Glass and M. L. Smith, "Meta-Analysis of Research on Class Size and Achievement," *Educational Evaluation and Policy Analysis,,* vol. 1, no. 1, pp. 2-16, 1979.

[15] C. M. Achilles, Let's Put Kids First, Finally: Getting Class Size Right., Thousand Oaks, CA: Corwin Press, Inc., 1999.

[16] K. H. Miles and L. Darling-Hammond, "Rethinking the Allocation of Teaching Resources: Some Lessons from High-Performing," *Educational Evaluation and Policy Analysis,,* vol. 20, no. 1, pp. 9-29, 1998.

[17] N. C. f. E. S. S. a. S. S. (. U.S. Department of Education, "Public School Teacher Data File," U.S. Department of Education, Washington, D.C., 2008.

[18] N. S. o. S. E. (NSSE), "NSSE About NSEE," National Survey of Student Engagement (NSSE), 2016. [Online]. Available: http://nsse.indiana.edu/html/about.cfm. [Accessed 12 December 2016].

[19] L. Gordon, "UC seeks to boost Californians' enrollment by 10,000 by 2018," Los Angeles Times, 9 November 2015. [Online]. Available: http://www.latimes.com/local/education/la-me-ln-uc-enroll-20151109-story.html. [Accessed 10 January 2016].

[20] K. Wilson and J. H. Korn, "Attention During Lectures: Beyond Ten Minutes," *Teaching of Psychology,* vol. 34, no. 2, pp. 85-89, 2007.

[21] B. Zimmerman, "Self-Efficacy: An Essential Motive to Learn," *Contemporary Educational Psychology,* vol. 25, no. 1, pp. 82-91, 2000.

[22] S. A. Jensen, "In-Class Versus Online Video Lectures : Similar Learning Outcomes, but a Preference for In-Class.," *Teaching of Psychology,* vol. 38, no. 4, pp. 298-302, 2011.

[23] M. Anteby and R. Khurana, "A New Vision - The Human Relations Movement - Baker Library | Bloomberg Center, Historical Collections," 30 June 2007. [Online]. Available: http://www.library.hbs.edu/hc/hawthorne/. [Accessed 21 August 2013].

[24] N. S. o. S. E. (NSSE), "NSSE Summary Tables," 2016. [Online]. Available: http://nsse.indiana.edu/html/summary_tables.cfm. [Accessed 13 December 2016].

[25] T. Bailey, J. Bashford, A. Boatman and e. al., Strategies for Postsecondary Students in Developmental Education: A Practice Guide for College and University Administrators, Advisors, and Faculty. NCEE 2017-4011, Washington, D.C.: What Works Clearinghouse, 2016.

[26] F. J. Roethlisberger, W. Dickson and H. A. Wright, Management and the Worker: an Account of a Research Program Conducted by the Western Electric Company, Hawthorne Works, Chicago., Cambridge: Harvard University Press, 1939.

[27] A. Maslow, "A Theory of Human Motivation," *Psychological Review,* vol. 50, no. 4, pp. 370-396, 1943.

[28] S. McLeod, "Maslow's Hierarchy of Needs - Simply Psychology," Simply Psychology, 2007. [Online]. Available: https://www.simplypsychology.org/maslow.html. [Accessed 22 October 2013].

[29] J. Valade, PHP 5 for Dummies, For Dummies, 2004.

[30] A. Silberschatz, H. F. Korth and S. Sudarshan, Database System Concepts, 6th ed., McGraw-Hill Education, 2010.

[31] WordPress, "Installing WordPress << WordPress Codex," 15 January 2016. [Online]. Available: https://codex.wordpress.org/Installing_WordPress.

[32] Moodle, "Installation quick guide - MoodleDocs," 15 January 2016. [Online]. Available: https://docs.moodle.org/30/en/Installation_quick_guide.

[33] R. Ramakrishnan and J. Gehrke, Database Management Systems, 3rd ed., McGraw-Hill Higher Education, 2002.

[34] R. H. Rockland, "Reducing the Information Overload: A Method on Helping Students Research Engineering Topics Using the Internet," *IEEE Transactions on Education,* vol. 43, no. 4, pp. 420-425, 2000.

[35] J. A. Hoxmeier and C. DiCesare, "System Response Time and User Satisfaction: An Experimental Study of Browser-based Applications," in *AMCIS 2000 Proceedings*, Long Beach, CA, 2000.

[36] F. Nah, "A Study on Tolerable Waiting Time: How Long are Web Users Willing to Wait?," *Behaviora and Information Technology,* vol. 23, no. 3, pp. 153-163, 2004.

[37] S. Deerwester, S. T. Dumais, G. W. Furnas and e. al., "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science,* vol. 41, no. 6, pp. 391-407, 1990.

[38] T. K. Landauer, P. W. Foltz and D. Laham, "An Introduction to Latent Semantic Analysis," *Discourse Processes,* vol. 25, pp. 259-284, 1998.

[39] P. W. Foltz, M. A. Britt and C. A. Perfetti, "Reasoning from Multiple Texts: An Automatic Analysis of Readers' Situation Models," in *Proceedings of the 18th Annual Cognitive Science*

Conference, Hillsdale, NJ, 1996.

[40] F. Vahid, Digital Design with RTL Design, VHDL, and Verilog Second Edition, Hoboken, NJ: John Wiley & Sons, Inc., 2010.

[41] S. L. Harris and D. M. Harris, Digital Design and Computer Architecture: ARM Edition 1st Edition, Morgan Kaufmann: Waltham, MA, 2015.

[42] M. Greenberg, Differential Equations and Linear Algebra, Upper Saddle River, NJ: Prentice Hall, 2000.

[43] D. A. Grossman and O. Frieder, Information Retrieval, Springer Netherlands, 2004.

[44] D. Hull, "Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing," in *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference*, Dublin, Ireland, 1994.

[45] V. Abedi, M. Yeasin and R. Zand, "Empirical Study Using Netork of Semantically Related Associations in Bridging the Knowledge Gap," *Journal of Translational Medicine,* vol. 12:324, 2014.

[46] P. J. Kwantes, "Using context to build semantics," *Psychonomic Bulletin & Review,* vol. 12, no. 4, pp. 703-710, 2005.

[47] I. University of California, "EEE Legacy," 1 May 2017. [Online]. Available: https://eee.uci.edu/. [Accessed 13 May 2017].

[48] D. o. T. a. Learning, "Center for Engaged Instruction - CEI at UC Irvine," 3 May 2017. [Online]. Available: http://cei.uci.edu/. [Accessed 3 May 2017].

[49] O. o. I. Technology, "Office of Information Technology - Serving UC Irvine's Technology Needs," 2017. [Online]. Available: http://www.oit.uci.edu/. [Accessed 3 May 2017].

[50] P. Dizikes, "Better wisdom from crowds | MIT News," 25 January 2017. [Online]. Available: http://news.mit.edu/2017/algorithm-better-wisdom-crowds-0125. [Accessed 18 May 2017].

[51] B. Aydin, Y. Yilmax, Y. Li and e. al., "Crowdsourcing for Multiple-Choice Question Answering," in *Proceedings of the Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence*, Quebec, Canada, 2014.

[52] R. Villa, J. Thousand, S. W. and S. Stainback, Restructuring for Caring and Effective Education, Baltimore: Paul Brookes, 1992.

[53] U. Regents, "University of California - Accountability Report," UC Regents, 2013. [Online]. Available: http://accountability.universityofcalifornia.edu/index/9.4. [Accessed 26 August 2013].

[54] U. Regents, "University of California - Accountability Report," UC Regents, 2009. [Online]. Available: http://accountability.universityofcalifornia.edu/2009/index/7.8. [Accessed 26 August 2013].

[55] E. Yudko, R. Hirokawa and R. Chi, "Attitudes, Beliefs, and Attendance in a Hybrid Course," *Computers & Education,* vol. 50, pp. 1217-1227, 2008.