Integrating Human and Machine Intelligence for Enhanced Curriculum Design

Shayan Doroudi

CMU-CS-19-110

May 15, 2019

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Emma Brunskill, Chair
Vincent Aleven
Ken Koedinger
Chinmay Kulkarni
Eric Horvitz (Microsoft Research)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This research was sponsored by the PIER Fellowship, Microsoft, Google, the Air Force Research Laboratory under grant number F87501720152, and the Department of Education under grant numbers R305A130215 and R305B150008. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: curriculum design, learnersourcing, content creation, reinforcement learning, instructional sequencing, student modeling, model misspecification, bias-variance tradeoff

If there is any good in these pages,
it is for Fatima and her father
and her husband
and her two sons,
peace and blessings be upon them.

Abstract

From the mechanical teaching machines of the early twentieth century to the wave of massive open online courses in recent years, many have been motivated by the dream of delivering a personalized adaptive curriculum to each learner. To achieve this dream, many researchers have focused on rule-based systems that rely on extensive domain expertise and psychological theories. While this approach has led to the development of successful intelligent tutoring systems with high quality content, (1) developing such systems can be very costly and (2) these systems typically employ a very limited form of adaptive instructional sequencing. In contrast, some researchers are now starting to apply black box machine learning algorithms to do adaptive instructional sequencing. However, these approaches have had relatively limited impact to date. Instead, I propose several techniques for impactful, cost-effective semi-automated curriculum design that combine machine learning, human computation, and principles from the learning sciences. My thesis will focus on two pieces of the curriculum design process: (1) content creation and curation and (2) instructional sequencing. First, I study the prospects of using learner-generated work for low-cost content creation. I explore this in the context of crowdsourcing tasks, where new kinds of work may require on-demand training. For two different kinds of crowdsourcing tasks, I show that learner-generated content can potentially be a useful way of teaching future learners, provided that the best content is automatically curated. Second, I show that due to model misspecification, relying on simple models of student learning can lead to making misinformed judgments about how to sequence content for students, including inequitable outcomes for low-performing students. To mitigate this problem, I suggest two ways in which we can effectively use models to perform instructional sequencing: (1) using multiple models of learning to develop instructional policies that are robust to how students actually learn, and (2) combining models of learning based on psychological theory with data-driven approaches. The broader theme of my thesis is that by integrating human and machine intelligence, we can improve upon efforts to better teach students in semi-automated ways.

Acknowledgments

I begin by thanking the One who provided me with *everything* that I needed to reach this point—physically, intellectually, and spiritually—and to whom I can never show enough gratitude:

> If I try my best and strive throughout all ages and all times—if I live them—to thank properly for only one of Your blessings, I will not be able to do that, except through Your favor, which would then require me to thank You again with an ever-lasting and new thanking.[1]

It was only with the help and support of many individuals that I was able to complete this thesis. I thank my advisor, Emma Brunskill, for all her support, guidance, and insights over the past six years. Emma provided much needed guidance and direction when I was lost and stuck in the weeds of research and was also very supportive of me pursuing my own interests and projects—even though many of my research directions were not typical for a machine learning research group. I was very fortunate to have an advisor who is not only an expert in machine learning but also passionate about education and doing impactful applied work.

I was also extremely fortunate to have had a thesis committee comprised of experts in machine learning, the learning sciences, and human-computer interaction. Vincent Aleven gave fruitful advice and mentorship from a learning sciences and intelligent tutoring systems lens that helped strengthen my work. Ken Koedinger provided much-needed advice and push-back to advance my work and ensure my thesis was authentic to the learning sciences and educational data mining literatures. Chinmay Kulkarni's pioneering work on peer interaction and learning at scale was inspirational for my own research, and I was fortunate to have his unique expertise and feedback on my thesis committee. Finally, having Eric Horvitz as one of my Microsoft Research internship mentors and as the external member on my committee was an amazing experience. Eric's visionary ideas and feedback helped push my work forward and were a constant source of inspiration. I am amazed as to the time and enthusiasm Eric could put into an intern's project, despite all of his other commitments.

Much of the work described in my thesis would literally not have been written without my collaborators. Vincent Aleven was a collaborator and advisor for the work described in Chapters 8 and 9, in addition to two papers which were not part of my dissertation. Ece Kamar was an excellent co-mentor for my Microsoft Research internship and long-term collaborator over the past five years. Ece provided me with day-to-day guidance and direction during my internship and she was always available to meet over the past few years. Chapters 3 and 4 were a result of my direct collaborations with Ece. Eric Horvitz was also a collaborator on the work described in Chapter 3. I am very grateful to Ece and Eric for allowing me to craft my own research project with their guidance.

I have also been fortunate to have collaborators on projects that were not included in my thesis. Ken Holstein was a collaborator on two of my early papers, but

[1] This is from the supplication recited by Imam Husayn ibn Ali on the Day of Arafah.

those projects were only the beginning of a long-term friendship and ongoing collaborations that will continue beyond my PhD. Conversations with Ken have helped shaped the work that I do, and his research is an inspiration for how to do impactful work in educational technology. I was also fortunate to have worked with Daniel Guo and Phil Thomas on theoretical machine learning projects. I have also had the great fortune of collaborating with Minsuk Chang, Juho Kim, and Keith Devlin on an ongoing collaboration, which was originally supposed to be part of my thesis. Finally, I thank everyone in Emma's lab at CMU and Stanford (the Artificial Intelligence for Human Impact lab) for their feedback and discussions that advanced my work.

I was blessed to have been part of the Program in Interdisciplinary Education Research (PIER), which provided me with financial support for two years and intellectual support for four years. David Klahr and Sharon Carver, the directors of the PIER program, were essentially *de facto* members of my thesis committee, and it was wonderful to benefit from their expertise in psychology and education. David and Sharon's PIER classes really helped me develop my own expertise in education research and the learning sciences, and provided with me with exemplars for how I want to develop my own courses going forward. I am immensely grateful to David, Sharon, and the entire PIER community for all their support and for helping me develop my identity as a learning scientist.

In the fourth year of PhD, I left Pittsburgh to be a Visiting Student Researcher at Stanford University. While it was unfortunate to physically leave the PIER community and all of the great assets that Carnegie Mellon provided, Stanford opened new doors for my research. In particular, the Lytics Lab has been a wonderful community with a diverse group of colleagues who have provided great feedback on my work. Petr Johanes, who played a pivotal role in running Lytics while I was there, has been a collaborator and friend. As with Ken Holstein, conversations with Petr have helped shaped the work I do; incidentally, several of those conversations have been between the three of us on an ongoing collaboration.

Aside from intellectual support, many individuals played a role in making my PhD as painless as possible. I thank Deb Cavlovich and Audrey Russo for handling all logistical matters for my work in the CS Department and PIER respectively. Several administrative assistants were also of great help, including Melissa Keaton (at CMU) and Tin Tin Wisniewski and Lashonda Renea Eagels (at Stanford). Finally, I thank the library staff at CMU and Stanford for helping me get all of the (at times obscure) books and articles that I needed.

I benefited greatly from the many friendships I formed during my PhD. I am particularly grateful to Majid Gholamzadeh Bazarbash and Ahmed Al-Khafaji for their friendship and help whenever I needed it. I am also grateful to the Imamia Organization of Pittsburgh, the Shia Association of the Bay Area, and the Virgin Mary Islamic Center for the spiritual and community support they provided.

I would not be who I am today without the constant support and encouragement of my family. I am indebted to my parents, Aladdin and Azar, in so many ways, both for all they have done for me before my PhD and for all they have done for me during this journey. I am also grateful to my aunts who have been there for me throughout the years. I am very thankful for my brother, Sherwin. He has been a role-model throughout my life, and my pursuing a PhD at CMU was in many ways following his footsteps and academic advice. Moreover, it was wonderful to have my brother and sister-in-law, Shirin, as family in Pittsburgh; they helped me in many ways and made my transition to Pittsburgh much easier. I also thank my extended family for supporting me from the beginning and my in-laws for accepting me into their family and being there for me over the past few years. Finally, I was immensely blessed to do my PhD at Carnegie Mellon, for it is where I met my wonderful wife, Iman. Her constant support, encouragement, and companionship over the past three years have made this journey so much better.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Over the past several years it has become increasingly clear to me, as to any thinking person today, that both psychology and the field of curriculum design itself suffer jointly from the lack of a theory of instruction... Let us, then, see whether we can set forth some possible theorems that might go into a theory of instruction.
>
> JEROME BRUNER, 1963

For decades, researchers, technologists, teachers, and students have been motivated by the dream of personalized, adaptive instruction for all students. In the 1920s-50s, attempts to realize this dream took the form of mechanical teaching machines that could give students step-by-step practice and feedback at an individualized pace [Ferster, 2014]. With the advent of computers and their emergence in research laboratories in the 1960s, researchers began creating computerized teaching machines that could sequence activities for students, at times using data-driven models of how students learn [Atkinson, 1972a,b, Smallwood, 1962]. With the formation of the field of artificial intelligence (AI), researchers in the 1970-80s began formulating theories about how experts learn to solve problems, which led to the development of intelligent tutoring systems (ITSs) that could guide students through problems in a step-by-step fashion, which persist to the present day [Anderson et al., 1985, Koedinger and Aleven, 2016]. Finally, with the rising popularity of machine learning, researchers have recently been turning to black box machine learning algorithms to automatically sequence curricula for students [Beck et al., 2000, Piech et al., 2015b, Reddy et al., 2017].

AI-based approaches to automating curriculum design can largely be classified into two types. Early approaches taken by cognitive scientists involved constructing rule-based systems that would encode domain experts' knowledge in the form of production rules that needed to be taught to the student. This approach has led to the development of many successful ITSs with high quality content. However, such systems can be very costly to build, with traditional estimates suggesting that building the content for each instructional hour on an ITS requires 200-300 hours of effort [Aleven et al., 2009, 2016a]. Although tools such as Cognitive Tutor Authoring Tools (CTAT) have been built to reduce the amount of time it takes to build such systems, they still require around 25 or more hours of a domain expert's time per instructional hour [Aleven et al., 2009]. Moreover, many ITSs have very limited forms of adaptive instructional sequencing, such as simply determining when a student is ready to move on to new skills [Corbett and Anderson, 1995]. In contrast, data-driven approaches using machine learning, which increasingly include black box algorithms like deep neural networks, can automatically infer how to sequence content for students from data. However, machine learning algorithms have a number of limitations—as I describe later in this thesis—that have seemingly limited their ability to adaptively sequence instruction for students. Machine learning methods have also been used for content creation, such as automatically creating quiz questions from natural language text [Heilman, 2011, Huang et al., 2014, Le et al., 2014] or automatically generating procedural problems for subjects such as algebra [Singh et al., 2012] and mathematical logic [Ahmed et al., 2013]. However, such techniques do not readily extend to complex, ill-defined tasks—especially tasks that computers cannot solve themselves.

In my thesis, I take the view that combining insights from both approaches rooted in human intelligence and approaches rooted in machine intelligence can help in automating various aspects of curriculum design, including *content creation*, *content curation*, and *instructional sequencing*. I demonstrate several techniques for impactful, cost-effective semi-automated curriculum design that combine machine learning, human computation, and principles from the learning sciences. First, I study the prospects of using learner-generated work for low-cost content generation. I explore this in the context of crowdsourcing tasks, where new kinds of work may require on-demand training. For two different kinds of crowdsourcing tasks, I show that learner-generated content can potentially be a useful way of teaching future learners, provided that the best content is automatically curated. Second, I show that due to model misspecification, relying on simple models of student learning can lead to making misinformed judgments about how to sequence content for students, including inequitable outcomes for low-performing students. To mitigate this problem, I suggest two ways in which we can effectively use models to perform instructional sequencing: (1) using multiple models of learning to develop instructional policies that are robust to how students actually learn, and (2) combining models of learning based on psychological theory with data-driven approaches.

My dissertation will be divided into two parts. The first part is focused on using learner-generated artifacts for low-cost content creation and using machine learned models to perform content curation over the learner-generated content. The second part is focused on combining theory-driven and data-driven approaches to instructional sequencing. Both approaches motivate the idea of integrating various forms of human intelligence (whether in the form of learner contributions or psychological theories) with machine intelligence to better automate curriculum design using educational technology. As such, I refer to the various processes I describe in this thesis as *semi-automated curriculum design*, to acknowledge the role that people play—in addition to machines—in the process of automating various pieces of curriculum design.

To be sure, prior work has also proposed several techniques that combine human and machine intelligence to improve curriculum design in educational technology [Koedinger et al., 2013]. For example, researchers have investigated how data-driven methods could automate the refinement of expert-defined cognitive models, which could in turn influence content design and instructional sequencing [Cen et al., 2006, Koedinger et al., 2012b]. Data-driven methods have also been used to automatically generate hints from prior student work [Barnes and Stamper, 2008]. Finally, interactive machine learning techniques (such as programming by demonstration) have been used to speed up the creation of intelligent tutoring systems, rather than relying on a human to explicitly author all of the production rules in an ITS [Blessing, 1997, Jarvis et al., 2004, Matsuda et al., 2015]. My thesis proposes several methods that fit into this space of integrating human and machine intelligence for semi-automated curriculum design.

However, in recent years, machine learning—especially deep learning—has made major advances in a variety of application areas, seemingly replacing the need for expert human intelligence (e.g., in the form of knowledge engineering or feature engineering). Therefore, as deep learning becomes increasingly popular, the need for human intelligence in automating curriculum design may also come into question. My thesis demonstrates, in two ways, why I believe there is a continued need for human intelligence in semi-automated curriculum design. First, on-demand content creation for ill-structured domains may become increasingly relevant as the future of work evolves due to advances in AI, but machines alone will likely not be able to create high quality content for such domains. Second, as I demonstrate below, machine learning algorithms used for instructional sequencing have a number of limitations that might be remedied by involving theories of learning and humans-in-the-loop to help decide how to sequence instruction for students.

Designing a curriculum, especially in the context of traditional education, is much broader than the aspects of curriculum that I tackle in my dissertation. First of all, the approaches to curriculum design that I consider here are only concerned with sequencing small scale pedagogical activities or generating worked examples for short tasks. By no means do I consider the automation of larger scale activities and content such as course projects or textbooks or aspects of curriculum that span across courses. Moreover, I do not consider automation with respect to other factors that significantly impact the curriculum, such as learning objectives, assessment, relationship to state standards, and the broader ecosystem in which the curriculum is positioned. Integrating these various aspects of curriculum design together simply speak to more places that currently require human intelligence in the process of semi-automated curriculum design and further motivate the broader approach that I advocate here of considering how people and machines can each contribute to different pieces of the curriculum design process.

The rest of the thesis is divided up as follows. Part I focuses on using learner-generated artifacts to create new low-cost educational content. This is particularly useful in contexts where expert examples are not readily available, such as on-demand training of crowdworkers to perform various kinds of complex tasks, which is the context of both sets of experiments that I run. Chapter 2 provides background on learnersourcing and related work that situates the contributions of the following two chapters. Chapter 3 describes experiments in the context of a complex web search task, where I test the efficacy of having crowd workers validate the work of their peers. In this case, my work suggests validating peer work can be as effective as, and possibly even more effective than, reading expert examples, provided that the peer solutions are sufficiently long. Chapter 4 describes experiments in the context of a more subjective task, where crowdworkers are asked to write reviews that compare pairs of similar products. Here, I test the efficacy of directly presenting peer-generated work as worked examples (rather than work to be validated). The experiments suggest that while randomly selected peer-generated examples do not seem to lead to learning gains on average, seeing high quality work can lead to improved performance on future tasks. Overall, these experiments show that peer-generated artifacts can be useful forms of training, but possibly only if the peer-generated work shown to students is algorithmically curated. My work preliminarily suggests that simple rules for choosing which peer-generated artifacts to present may be enough to find ones that are pedagogically effective, but future work should look into more sophisticated data-driven ways of curating the best peer content.

Part II considers an entirely different aspect of technology-driven curriculum design: how to effectively sequence content to improve student learning. Chapter 5 gives background on two common approaches to automated instructional sequencing that I focus on: cognitive mastery learning and reinforcement learning. In Chapter 6, I examine some limitations of the Bayesian knowledge tracing (BKT) model, a commonly used model to implement cognitive mastery learning in ITSs. In particular, using simulation studies, I show potential consequences of the model's bias in terms of outcomes for student learning. In Chapter 7, I further investigate how the bias of BKT could lead to inequitable outcomes for low-performing students. In Chapter 8, I turn to broader approaches to instructional sequencing using reinforcement learning and motivate mitigating the bias of relying on a single model of student learning by forming instructional policies that are robust to *multiple* models of student learning. Finally, in Chapter 9, to better understand how we can use data-driven approaches for instructional sequencing, I comprehensively review the empirical literature in this area. One of the conclusions of this review is that prior data-driven attempts at instructional sequencing have been most effective when they relied on psychological theories of learning. Much of the work described in Part II was motivated by a reflection on our own failed attempts to substantially improve student learning using data-driven instructional policies, which I claim was due to not finding a good way to approach the bias-variance tradeoff in instructional sequencing. These chapters provide pointers for ways to avoid the challenges we faced.

Chapter 10 concludes the paper by providing a summary of the key contributions of my thesis and presenting my thoughts on various ways in which we can integrate human intelligence—whether in the form of learning theories, learner contributions, or a teacher's guidance—with machine intelligence to more effectively automate curriculum design. As data-driven approaches become more and more popular in the automation of curricula and in the design of educational technology more broadly, I believe we need to take a step back and critically examine the variety of attempts that have been made to automate curriculum design to better understand what works and what does not. Data-driven methods certainly have a place in educational technology, but the question is how we can make the best use of them. I contend that data-driven methods will be most useful if we can discover how to best integrate them with people's ideas, contributions, and theories. Ultimately, only time will tell how educational technology can have the greatest impact. However, it is our responsibility to take a thoughtful approach, and my hope is that this thesis adds to the conversation of how we might take such an approach towards the goal of providing an adaptive personalized education for all.

Part I

Content Creation and Curation

Chapter 2

Background: Learnersourcing

> Through dialogue, the teacher-of-the-students and the students-of-the-teacher cease to exist and a new term emerges: teacher-student with students-teachers. The teacher is no longer merely the-one-who-teaches, but one who is himself taught in dialogue with the students, who in turn while being taught also teach. They become jointly responsible for a process in which all grow.
>
> Paulo Freire, 1968

In new online learning environments that attract large numbers of people, many learners either individually or collectively make artifacts in the course of their interactions with the learning environment. These learner-generated artifacts can potentially be used to impact the learning opportunities of future learners, via *learnersourcing* [Kim et al., 2015]. For example, in massive open online courses, students create open-ended artifacts such as essays, computer programs, designs, and mathematical proofs. These learner-generated artifacts are often presented to other learners in peer-evaluation exercises, and while the primary purpose of this is to scale grading [Piech et al., 2013], some instructors have treated evaluating peer work as an explicit learning opportunity [Devlin, 2013]. In Scratch, the popular online programming community and learning enviornment for kids, learners are encouraged to share their programs and remix other learners' programs [Resnick et al., 2009], which has been shown to serve as a pathway for learning [Dasgupta et al., 2016]. Finally, in crowdsourcing platforms, many crowdworkers do large numbers of tasks for requesters. While a lot of crowdsourcing tasks on websites like Amazon Mechanical Turk are microtasks, which are relatively easy, do not require creativity, and require little training, recent research has investigated crowdsourcing more complex work [Doroudi et al., 2016, Kittur et al., 2013b, Steuer et al., 2017]. For such tasks, workers might generate complex artifacts (e.g., product reviews or website designs). As such, these artifacts could be presented to other workers as an inspiration or means of better understanding how to perform the task.

Thus, one way in which *naturally* generated learnersourced artifacts can be used is to bootstrap the creation of low-cost curricula where we might not have the tools or time to create a high quality curriculum from scratch. Crowdsourcing is a particularly interesting domain to explore the effects of peer-generated artifacts for learning, because crowdsourcing tasks often come from ill-structured domains, where we do not have existing curricula to help teach workers. Moreover, the types of complex crowdsourcing tasks could be evolving over time as the future of work evolves and requesters have new needs. As such, I believe finding ways to automatically train crowd workers has implications for the future of work, where workers flock towards complex on-demand tasks and are in need of real-time, quick training.

In this part of my thesis, I examine how to perform this low-cost content generation for complex crowdsourcing tasks. I investigate two broad questions: (1) can peer-generated artifacts serve as a means to bootstrap the creation of viable content, and (2) how can we curate the best peer-generated content? Chapter 3 and Chapter 4 explore these two questions for two different kinds of tasks and two different ways of utilizing learnersourced work. Chapter 3 focuses on the efficacy of having workers validate peer-generated work in the context of a complex problem solving task, namely web search. I compare validating peer work with other modes of training, such as solving more tasks and reading an expert example. Chapter 4 examines the efficacy of using learner-generated work as examples to be read by crowdworkers, rather than validated in the context of a more subjective complex crowdsourcing task, namely peer reviews. In this context, I was interested in asking what kinds of peer-generated artifacts (i.e., single worked example, pair of worked examples, or learner-generated task guidelines) are effective when simply presented to crowd workers as resources.

In both cases, I find that peer-generated content can be an effective way of training crowdworkers, but possibly only if we present good content. Moreover, in both cases, I show that a simple rule for determining which content is most pedagogically useful (e.g., presenting peer work that is sufficiently long or of sufficiently high quality) can be an effective way of curating content. Even though these rules are simple, machine learning techniques can help in automatically discovering them. For example, in the web search domain (Chapter 3), I used a regression model to automatically discover what features of peer solutions make them pedagogically valuable, which found that solution length (and only solution length) was a good predictor of the future performance of workers who validate the given solution. For the work described in Chapter 4, I am currently investigating the use of machine learning models to predict the pedagogical value of peer-generated examples. I posit that more data-driven algorithms can help in refining the content curation process.

Using peer-generated content to help future learners, and especially crowdworkers, is an interdisciplinary endeavor, and our work draws on literature from various works in the learning sciences and crowdsourcing literatures, and especially in the intersection of these fields. In the remainder of this chapter, I will discuss the variety of related work that the work in this part of my thesis draws upon and contributes to.

2.1 Learnersourcing

The concept of *learnersourcing* refers to using work done by crowds of learners to help improve the educational experience of future learners [Kim et al., 2015]. Of most relevance to the present work are studies that have specifically looked at how to use learnersourcing to create new educational content that can help future learners [Farasat et al., 2017, Glassman et al., 2016, Heffernan et al., 2016, Mitros, 2015, Whitehill and Seltzer, 2017, Williams et al., 2016].

For example, Williams et al. developed a system called AXIS that had learners generate explanations to math word problems that could later be used to help other learners [Williams et al., 2016]. They used multi-armed bandits to automatically discover the explanations that learners found to be most useful. In a randomized experiment, they showed that learner-generated explanations that AXIS chose to present to students led to higher learning gains than explanations that did not meet a set of pre-specified quality checks. Their result is similar to ours in that it shows that not all learnersourced explanations are useful, but identifying good peer-generated explanations can be effective. However, they only compared their AXIS-chosen explanations to ones that were specifically thought to be bad, so it is not clear how effective random (or even above-median) explanations would have been. In my studies, I show that random peer-generated content is not necessarily effective, but either long peer work (Chapter 3) or high quality examples (Chapter 4) can be effective in improving workers' performance.

Similarly, Aleahmad et al. [2009] looked into crowdsourcing content creation to teachers and amateurs on the web who could create solutions to a Pythagorean theorem problem. They found that they could generate hundreds of high quality solutions (as measured via expert ratings) at a low cost and could automatically detect many of the poor solutions before having experts rate the solutions. However, the authors did not measure how much students actually learned from these solutions or how they compared to expert examples. More recently, Whitehill and Seltzer [2017] showed that crowdworkers could generate videos to teach logarithms at a cost of $5 per video, which had positive learning gains that were comparable to watching a Khan Academy video on logarithms. However, that both of these studies are not technically regarded as learnersourcing since they rely on non-learners external to the learning environment to create content for learners.

Moreover, all of these studies and others [Farasat et al., 2017, Glassman et al., 2016, Mitros, 2015] actively ask the crowd to create new content, which could be used to help future learners. This is referred to as *active learnersourcing* [Kim et al., 2015]. In this thesis, I primarily study using *passive learnersourcing* [Kim et al., 2015] in order to leverage artifacts that would naturally be generated by learners regardless of their role in helping other learners.[1] In a sense, I am interested in how to make more efficient use of work being done in learning and work environments that would traditionally not be seen as pedagogically valuable. Passive learnersourcing has the advantage of not requiring additional work or cost to create curricula, which could be particularly useful in crowdsourcing settings where requesters have a limited budget.

[1] One exception to this is that in Chapter 4, I study using peer-generated guidelines, which has to be actively elicited, however, our results do not indicate that they are necessarily more effective than just using examples, so I do not pursue this idea further.

More broadly, in recent years, researchers have written vision papers on how human computation can impact the future of education. Weld et al. [2012] described how human computation or crowdsourcing can address new challenges in personalizing online education in the wake of Massive Open Online Courses (MOOCs). One of the challenges the researchers discussed was content creation and curation in online courses, and how crowds of students could be used for that purpose. Their paper could be seen as a call to action for human computation researchers; this part of my dissertation can be seen as an answer to that call. Moreover, in Heffernan et al. [2016] predicted that "in many ways, the next 25 years of adaptive learning technologies will be driven by the crowd" and described their efforts to begin to use crowdsourcing for content creation in ASSISTments (a system that teachers use to teach mathematics in the classroom).

2.2 Peer Review

Reading and validating peer work is related to the literature on peer review in classroom settings and peer grading in MOOCs. Much of the research in this area has focused on either how to effectively use peer grading to scale assessment in large-scale online classes [Kulkarni et al., 2015, Piech et al., 2013] or on how peer review and feedback can benefit the receiver of the feedback [Dow et al., 2012, Falchikov, 1995, Gielen et al., 2010]. However, there is a growing body of research on the effects of peer review on the reviewer (or the giver of feedback). Sadler and Good [2006] studied how grading either one's own tests or one's peers' tests improve subsequent performance when re-taking the same test (after a week). They found that grading one's own test to be beneficial, but grading peer tests did not seem to improve the students' scores on the subsequent test. This may be because students can find their own mistakes when grading their own tests. Their inability to learn from grading peer tests may also be because they were simply grading and not providing any feedback. Wooley et al. [2008] found evidence for this second hypothesis by finding that college students did not write better papers after simply grading their peers' papers, but that students who were asked to also give feedback wrote better papers than students who did not review peer work. This suggests the importance of giving feedback or at least requiring students to engage with peer work in more effortful ways. Consistent with this, Cho and MacArthur [2011] found that reviewing peer papers led to greater writing quality on a later writing assignment than simply reading peer papers or not engaging with peer papers at all. Moreover, Lundstrom and Baker [2009] found that giving feedback in a second language writing task led to greater improvements than receiving peer feedback in future writing tasks throughout the course. All in all, peer review is likely an effective way of engaging with learner-generated content, because it requires actively engaging with content rather than passively engaging with it. There is a wide body of learning sciences literature that supports active engagement [see e.g., Chi and Wylie, 2014].

While I build on this work, my work differs from prior work on peer review in a number of ways. First of all, prior work does not view peer-generated content as content per se, but rather looks at the process of peer review as a well-established practice that is used in educational settings. By viewing peer solutions as content, I am, interested in seeing how engaging with such content compares with reading expert examples, and I am interested in alternative ways of engaging with peer-generated content beyond just grading them. For example, in Chapter 4 I test the efficacy of using learner-generated solutions as worked examples. In short, I am not tied to the process of peer review, although I do analyze its efficacy in teaching learners in Chapter 3. Second of all, this prior work does not look at how to curate the best content. In the following two chapters, I observe that reviewing random peer work is not necessarily effective, and so content curation is necessary. In the traditional classroom and MOOC context, typically instructors would like to have all work reviewed, and so content curation is not a concern. Of course, in the crowdsourcing context, we may also need to have all work reviewed, so it may be necessary to distinguish reviewing for pedagogical purposes and reviewing for grading purposes. Exploring this tradeoff in both crowdsourcing contexts and traditional educational contexts would be interesting to explore in future work. Furthermore, reviewing pedagogically valuable peer work may also better prepare learners for future peer review tasks that are necessary for grading.

2.3 Crowd Training

Several prior studies explore the training of crowdworkers [Dontcheva et al., 2014, Oleson et al., 2011, Singla et al., 2014, Zhu et al., 2014]. Oleson et al. proposed the use of gold standards as a form of training on relatively simple microtasks, but their primary focus was on the use of gold standards for quality assurance rather than on quantifying their efficacy in training [Oleson et al., 2011]. Willett et al. used examples for training workers and for calibrating their work to match the requesters' expectations on visualization microtasks and found that workers exposed to examples generated higher quality responses than workers who did not [Willett et al., 2012]. Similarly, Mitra et al. used examples followed by a qualification test and found that this training improved the quality of workers' data annotations [Mitra et al., 2015]. Singla et al. used machine learning to optimize which training examples to show workers in simple classification tasks [Singla et al., 2014]. Moving beyond microtasks, Dontcheva et al. proposed constructing platforms that integrate training and crowdsourcing in a photo editing environment [Dontcheva et al., 2014]. The Duolingo system[2] similarly combines language learning and a crowdsourced translation service in a single platform. However, the construction of such platforms requires domain-specific knowledge and engineering and can be quite costly to build. Dow et al. [2012] showed that either having workers self-assess their product reviews or having experts give feedback on their product reviews improves the quality of subsequent reviews. Of most relevance to our work, Zhu et al. [2014] compared two forms of training. They found that reviewing the work of other workers is a more effective form of training than doing more tasks. This comparison is similar to our first experiment in Chapter 3 and could be considered as advocating for a form of passive learnersourcing; however, the tasks they studied were subjective tasks (e.g., brainstorming novel ideas) that required creativity rather than strategy-driven complex problem solving tasks that have objective answers [Zhu et al., 2014].

[2] www.duolingo.com

2.4 Crowdsourcing as Learning at Scale

In addition to work on learnersourcing and work related to training crowd workers, there is an emerging body of work studying learning in crowdsourcing platforms. Recent work has looked into understanding how crowdsourcing platforms can support learning as part of crowdwork and to foster the longer term development of worker skills [Dontcheva et al., 2014, Jun et al., 2018, Krause et al., 2016, Suzuki et al., 2016]. For example, Jun et al. showed that workers value learning about scientific studies that they participate in [Jun et al., 2018]. Our work fits into this narrative of crowdsourcing platforms as not just platforms to test learning at scale ideas, but to *enact and support* learning at scale.

2.5 Learning Sciences

To develop hypotheses about different forms of training, I turn to the learning sciences literature, where instructional interventions have been more intensively studied than in the crowdsourcing community. *Worked examples*, or expert step-by-step solutions to a task, have been shown to be an effective form of teaching [Salden et al., 2010b, VanLehn, 1996]. Research has shown the presence of the *worked example effect*: reviewing examples is more effective than solving the tasks for learning, at least for novices [Sweller and Cooper, 1985]. While the *expertise reversal effect* claims that for more advanced students the opposite is true—solving problems is more effective than reviewing examples [Kalyuga et al., 2001]—more recent work demonstrated that in a less-structured domain, the worked example effect holds for both novices and advanced students [Nievelstein et al., 2013]. This finding may be relevant to complex problem solving tasks, such as complex web search, as they are less-structured than problems in many typical educational settings. Additionally, learning sciences research has shown that novices learn more from their peers than from experts when being trained directly on the task they are tested on [Hinds et al., 2001]. However, expert examples have been shown to be more effective than peer examples on transfer tasks—tasks that share some, but not all, properties of the examples [Boekhout et al., 2010, Hinds et al., 2001, Lachner and Nückles, 2015]. In both crowdsourcing domains below, I look at tasks that are very different from one another, so we expect many of our tasks to be in the transfer regime. I aim to explore how these results generalize to the crowdsourcing of complex tasks.

2.6 Ill-Structured Domains

An ill-structured domain is a domain where the problem space is not (or cannot be) entirely well-specified and as such we do not have algorithms that can solve problems from such a domain [Newell, 1969]. According to Newell [1969], an ill-structured problem is one which humans can often solve but known algorithms (at least under the current state-of-the-art) cannot. It is thus natural that crowdsourcing tasks are often from ill-structured domains, because the very reason we resort to solving them with people is that we cannot do so with computers. As such, they can also be difficult to teach.

Prior work in educational psychology and the learning sciences has explored instruction for particular ill-structured domains. Some work has shown that examples can help novice learners on retention and near transfer tasks (i.e., tasks that share similar features with the example), but less so on far transfer tasks [Kyun et al., 2013]. Other work has suggested that direct instruction (including giving an example for a single task) is *in principle* not beneficial for ill-structured domains [Spiro and DeSchryver, 2009]. Instead, researchers have suggested using multiple examples from different tasks in ill-structured domains [Spiro et al., 1988], so learners can understand the variety of distinct cases that fall into that domain. This work motivates the methods we test for training crowdworkers. However, the present work expands on this literature on ill-structured domains by exploring the use of *peer-generated* work for teaching in ill-structured domains.

Chapter 3

Validating Peer Work*

> The peer approach requires students to take control of the writing process, and to learn to critique their own work as they review the work of other students. This may be the most important underlying idea: students must become active participants in their own learning, not passive recipients of sacred knowledge from an authoritative outside source.
>
> Helen Schneider, 1988

To date, crowdsourcing has largely focused on tasks that can be solved without special training or knowledge. However, many interesting tasks cannot be solved effectively by untrained crowdworkers. Examples of such tasks include using web search to answer complicated queries, designing an itinerary for someone going on a vacation [Zhang et al., 2012], and condensing an academic article to an accessible summary for the general public [Kittur et al., 2011]. One approach to crowdsourcing such tasks is to decompose them into smaller subtasks that are easier to solve [Bernstein et al., 2010, Cheng et al., 2015, Kittur et al., 2011, 2013a, Zhang et al., 2012]. However, such task decomposition frequently requires the careful design and engineering of task-specific workflows. We investigate the less-studied case of crowdsourcing tasks that cannot be decomposed in a straightforward manner. Specifically, we consider the class of **complex problem solving tasks** that satisfy the following three properties: (1) there is a large space of potential strategies that workers can use to solve the tasks, (2) workers have the capacity to solve the tasks by discovering and trying different strategies, and yet (3) a significant proportion of untrained workers are unable to solve these tasks with high accuracy. In this chapter we look at complex web search tasks, where workers have to perform a series of web search queries to find the right answer, as a prototypical type of complex problem solving task.

Little is known about how to optimally train crowdworkers to perform complex tasks in a cost-effective way. Experts may be unavailable or unwilling to invest time into training crowdworkers and, in many cases, requesters themselves do not understand how to solve their complex tasks let

*This chapter was adapted from Doroudi et al. [2016].

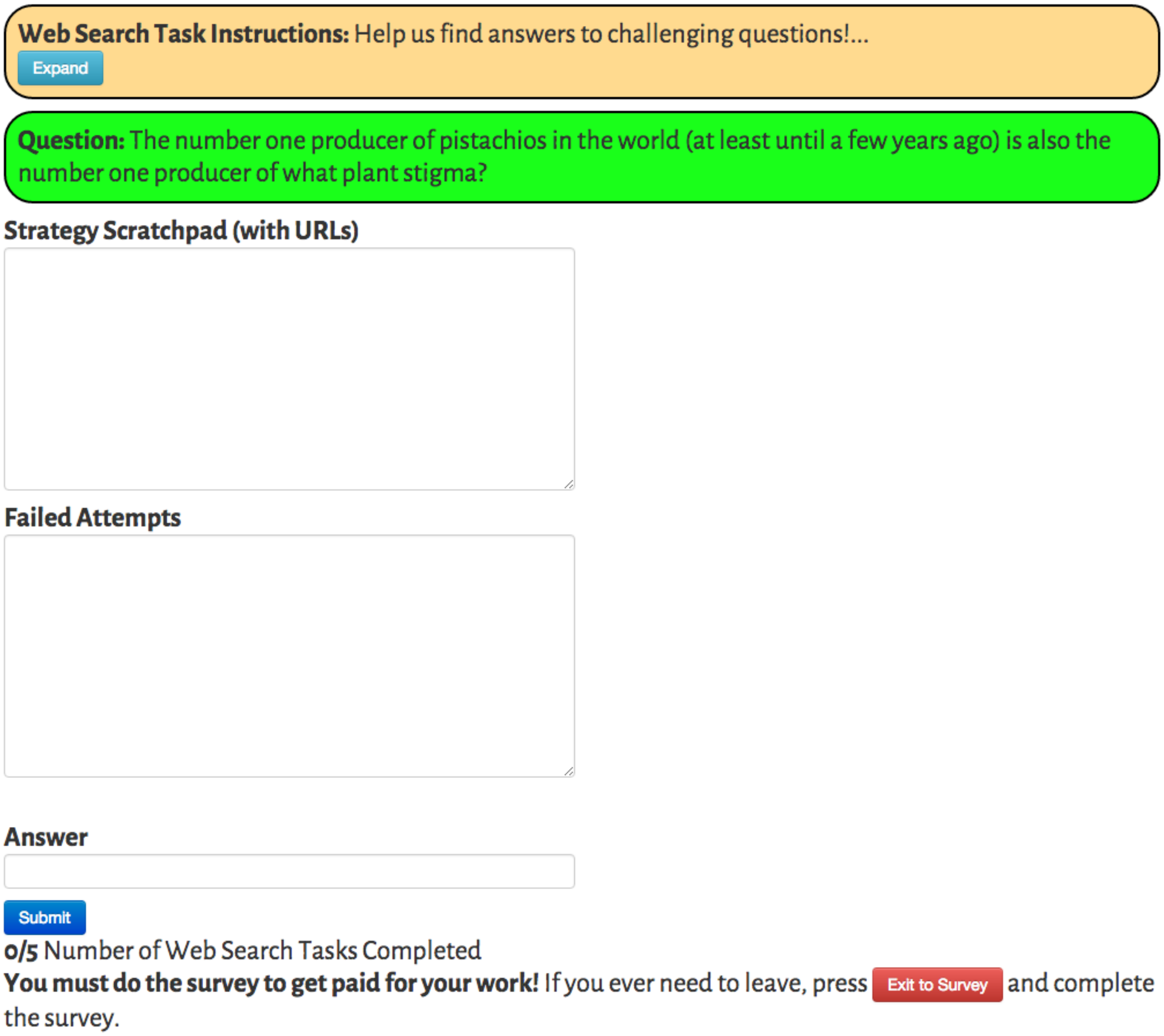
alone how to train others to solve them. Furthermore, there may be a large continuum of possible strategies for solving these problems, with different strategies being optimal in different instances of the task. The strategies used to solve the task may also need to change over time (e.g, to detect web spam, workers need to adapt to adversarial shifts in spammer strategies over time). As such, it can be unwieldy, if not impossible, to write a comprehensive a set of standing instructions on how to approach these tasks. This makes recent peer-generated work an attractive potential source of content for training crowd workers. But is such content pedagogically effective? This is the key question we hope to answer in this chapter.

We ran two experiments that explore how to train crowdworkers to do complex web search tasks, with a focus on how effective peer-generated solutions might be for training. The first experiment compares various forms of training including expert examples, learning by doing, and validating peer work. From the first experiment, we find that expert examples appear to be the most effective form of training but validating peer work also appears to possibly effective. We then use a regression model to predict which peer solutions are most pedagogically effective, and use that to inform the design of our second experiment. In the second experiment, we found that presenting workers with crowdsourced solutions that were filtered by length results in as high learning gains as using expert examples, and possibly even higher learning gains if at least one very long solution is validated. These results highlight the feasibility of developing automated training pipelines for crowdsourcing complex tasks that run in the absence of domain expertise.

3.1 Task Design

Complex web search is an interesting domain for crowdsourcing because the desired answer cannot typically be captured by simply querying a search engine once. Instead, workers need to explore and aggregate multiple sources of information to reach an answer [Aula and Russell, 2008]. Furthermore, complex web search is a prototypical member of the class of complex problem solving tasks that we defined above; users utilize a variety of different strategies when approaching web search problems [Thatcher, 2008], and as we show below, on average, untrained workers solve the web search tasks with 50% accuracy, indicating that workers do have the capacity to solve these tasks, but without training, many solve them with low accuracy.

I developed a pool of questions that were designed to typically require several searches to find the right answer. Questions were adapted and influenced from search tasks given in agoogleaday.com since these questions were found to be at the appropriate level of complexity. Figure 3.2 shows one such question along with an expert solution that we wrote. The rest of the questions are shown in Appendix A. We ran a pilot study to decide how many questions to show in each training session. We hypothesized that using too many training questions may decrease worker engagement with the study while using too few questions may decrease the effectiveness of training. After trying training sessions with one, two, and three training tasks, we found that some workers found it unreasonable to have to review three expert examples before being able to start the task. We settled on giving workers two training tasks. We refer to the two training



Web Search Task Instructions: Help us find answers to challenging questions!...

Question: The number one producer of pistachios in the world (at least until a few years ago) is also the number one producer of what plant stigma?

Strategy Scratchpad (with URLs)

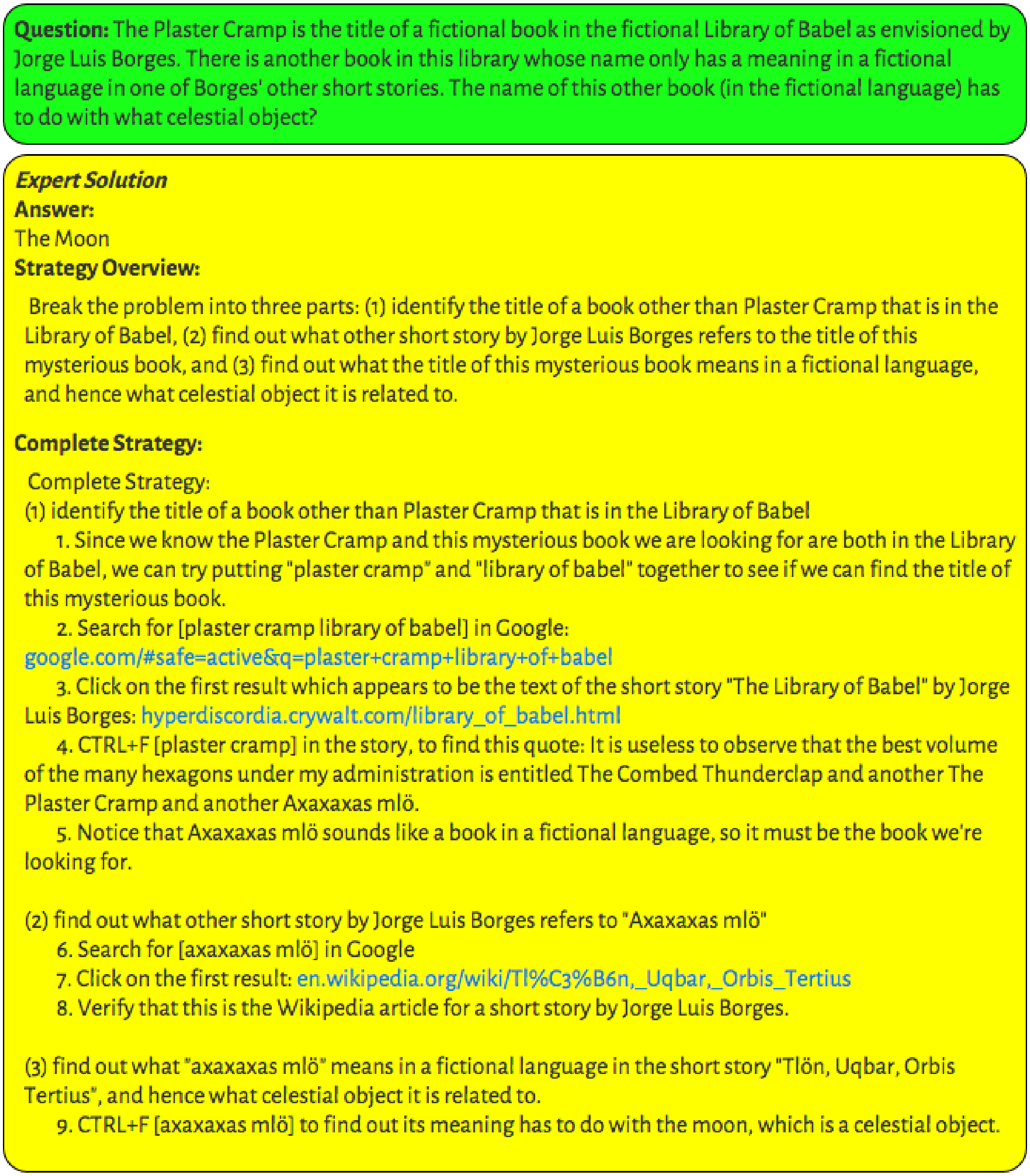
Failed Attempts

Answer

0/5 Number of Web Search Tasks Completed

You must do the survey to get paid for your work! If you ever need to leave, press Exit to Survey and complete the survey.

Figure 3.1: Web search task interface



Question: The Plaster Cramp is the title of a fictional book in the fictional Library of Babel as envisioned by Jorge Luis Borges. There is another book in this library whose name only has a meaning in a fictional language in one of Borges' other short stories. The name of this other book (in the fictional language) has to do with what celestial object?

Expert Solution

Answer:

The Moon

Strategy Overview:

Break the problem into three parts: (1) identify the title of a book other than Plaster Cramp that is in the Library of Babel, (2) find out what other short story by Jorge Luis Borges refers to the title of this mysterious book, and (3) find out what the title of this mysterious book means in a fictional language, and hence what celestial object it is related to.

Complete Strategy:

Complete Strategy:

(1) identify the title of a book other than Plaster Cramp that is in the Library of Babel

1. Since we know the Plaster Cramp and this mysterious book we are looking for are both in the Library of Babel, we can try putting "plaster cramp" and "library of babel" together to see if we can find the title of this mysterious book.

2. Search for [plaster cramp library of babel] in Google: google.com/#safe=active&q=plaster+cramp+library+of+babel

3. Click on the first result which appears to be the text of the short story "The Library of Babel" by Jorge Luis Borges: hyperdiscordia.crywalt.com/library_of_babel.html

4. CTRL+F [plaster cramp] in the story, to find this quote: It is useless to observe that the best volume of the many hexagons under my administration is entitled The Combed Thunderclap and another The Plaster Cramp and another Axaxaxas mlö.

5. Notice that Axaxaxas mlö sounds like a book in a fictional language, so it must be the book we're looking for.

(2) find out what other short story by Jorge Luis Borges refers to "Axaxaxas mlö"

6. Search for [axaxaxas mlö] in Google

7. Click on the first result: en.wikipedia.org/wiki/Tl%C3%B6n,_Uqbar,_Orbis_Tertius

8. Verify that this is the Wikipedia article for a short story by Jorge Luis Borges.

(3) find out what "axaxaxas mlö" means in a fictional language in the short story "Tlön, Uqbar, Orbis Tertius", and hence what celestial object it is related to.

9. CTRL+F [axaxaxas mlö] to find out its meaning has to do with the moon, which is a celestial object.

Figure 3.2: Expert example for training Question Y

questions as X and Y, and we refer to the five test questions that we give workers as A, B, C, D, and E. We note that optimizing the quantity of training is an interesting question that we do not further explore here.

In the web search tasks, workers were instructed both to provide an answer to the question and to write down their thought process and record each step they took towards the answer (including all visited URLs) in a web form that we call the **strategy scratchpad**. Workers were also asked to record unsuccessful strategies in what we call the **failed attempts box**. Figure 3.1 shows the main interface that workers interacted with when completing the web search tasks. An example of a worker's solution is shown at the top of Figure 3.3. In this particular solution, we see that despite having many failed attempts, the worker eventually found the correct answer using a strategy that was drastically different from the expert example (and from other workers).

3.2 Hypotheses

We formulated several hypotheses on the efficacy of various forms of training based on the prior findings in the literature. First, the worked example effect [Sweller and Cooper, 1985] suggests the following hypothesis:

Hypothesis 1 *Reviewing expert examples is an effective form of training in terms of increasing the accuracy of workers in finding answers to complex search tasks.*

Second, Zhu et al. [2014] showed that reviewing the work of peer workers provides more effective training than doing more tasks. This can be seen as an analogue to the worked example effect, but instead of simply reading through an example, the worker must read *and* validate the work of a peer worker. However, the learning sciences literature suggests that expert examples are more effective than peer examples for transfer tasks [Boekhout et al., 2010, Hinds et al., 2001, Lachner and Nückles, 2015]. These findings suggest the following hypothesis:

Hypothesis 2 *Validating other crowdworkers' solutions is also beneficial for increasing worker accuracy but less so than reviewing expert examples.*

Similarly we hypothesize that validating high-quality peer solutions, which are similar to expert solutions, will lead to more effective training than validating low-quality solutions. Furthermore, we might imagine that the validation process has a benefit beyond simply reading through an example, so the training benefit from validating such high quality peer solutions may even exceed that of reviewing expert examples. These hypotheses can be formulated as follows:

Hypothesis 3 *Having workers validate filtered crowdsourced solutions that are higher quality than average leads to a greater increase in accuracy than having them review unfiltered solutions.*

Hypothesis 4 *If the solutions presented to workers are of high enough quality, this will have at least the same effect as presenting workers with expert examples.*

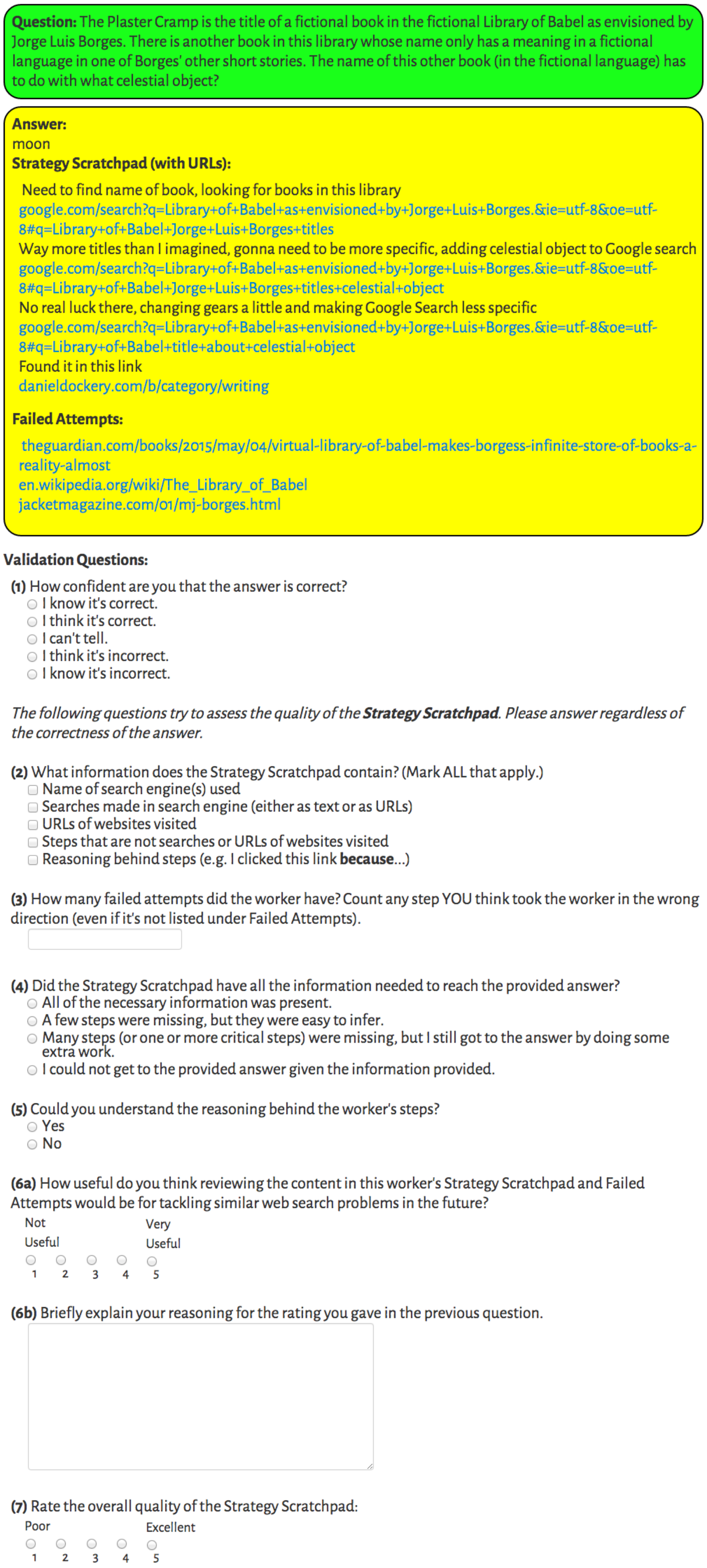
Confirming these hypotheses would provide support for building domain-agnostic pipelines that train crowdworkers using their peers' work. Such pipelines could improve the quality of training over time via methods for presenting the best peer solutions to workers. Eventually, such a pipeline could accrue a repository of high quality worked examples from crowd work without requiring the requester to have extensive domain knowledge. Such a pipeline would have the additional benefit of providing quality control of work performed on complex tasks via peer validation.

3.3 Experimental Design

We ran all of our experiments on Amazon Mechanical Turk.[1] Workers were assigned to one of several different training conditions (i.e. five in Experiment I and three in Experiment II) as soon as they accepted our Mechanical Turk Human Intelligence Task (HIT)[2]. The workers were assigned to the conditions in a round robin fashion to balance the number of workers assigned to each condition. Workers were first presented with an informed consent form that did not reveal we were studying worker training. Upon providing consent, workers were presented with condition specific instructions followed by two training tasks (unless they were in the control condition), possibly an additional set of instructions depending on the condition, and then five test tasks. For both training and test questions, we assigned the questions to workers in a random order. For example, workers were as likely to see training question X and then Y as they were to see Y and then X. While doing any of the tasks, the worker could choose to stop working on the HIT by completing an exit survey, which was required for payment. When workers began the survey, we revealed that the primary purpose of the study was to analyze the efficacy of various forms of training, and asked them several questions about the tasks in general and about the efficacy of the training they received in particular.

3.4 Experiment I

The first experiment was performed to compare various forms of training inspired by the literature. We sought to find the most effective method for training as characterized by several metrics including worker accuracy. We focused on validating Hypotheses 1 and 2 on exploring the relative efficacies of workers reviewing expert examples and validating peer-generated solutions.



Question: The Plaster Cramp is the title of a fictional book in the fictional Library of Babel as envisioned by Jorge Luis Borges. There is another book in this library whose name only has a meaning in a fictional language in one of Borges' other short stories. The name of this other book (in the fictional language) has to do with what celestial object?

Answer:
moon
Strategy Scratchpad (with URLs):

Need to find name of book, looking for books in this library
google.com/search?q=Library+of+Babel+as+envisioned+by+Jorge+Luis+Borges.&ie=utf-8&oe=utf-8#q=Library+of+Babel+Jorge+Luis+Borges+titles
Way more titles than I imagined, gonna need to be more specific, adding celestial object to Google search
google.com/search?q=Library+of+Babel+as+envisioned+by+Jorge+Luis+Borges.&ie=utf-8&oe=utf-8#q=Library+of+Babel+Jorge+Luis+Borges+titles+celestial+object
No real luck there, changing gears a little and making Google Search less specific
google.com/search?q=Library+of+Babel+as+envisioned+by+Jorge+Luis+Borges.&ie=utf-8&oe=utf-8#q=Library+of+Babel+title+about+celestial+object
Found it in this link
danieldockery.com/b/category/writing

Failed Attempts:

theguardian.com/books/2015/may/04/virtual-library-of-babel-makes-borgess-infinite-store-of-books-a-reality-almost
en.wikipedia.org/wiki/The_Library_of_Babel
jacketmagazine.com/01/mj-borges.html

Validation Questions:

(1) How confident are you that the answer is correct?
○ I know it's correct.
○ I think it's correct.
○ I can't tell.
○ I think it's incorrect.
○ I know it's incorrect.

The following questions try to assess the quality of the ***Strategy Scratchpad****. Please answer regardless of the correctness of the answer.*

(2) What information does the Strategy Scratchpad contain? (Mark ALL that apply.)
□ Name of search engine(s) used
□ Searches made in search engine (either as text or as URLs)
□ URLs of websites visited
□ Steps that are not searches or URLs of websites visited
□ Reasoning behind steps (e.g. I clicked this link **because**...)

(3) How many failed attempts did the worker have? Count any step YOU think took the worker in the wrong direction (even if it's not listed under Failed Attempts).

(4) Did the Strategy Scratchpad have all the information needed to reach the provided answer?
○ All of the necessary information was present.
○ A few steps were missing, but they were easy to infer.
○ Many steps (or one or more critical steps) were missing, but I still got to the answer by doing some extra work.
○ I could not get to the provided answer given the information provided.

(5) Could you understand the reasoning behind the worker's steps?
○ Yes
○ No

(6a) How useful do you think reviewing the content in this worker's Strategy Scratchpad and Failed Attempts would be for tackling similar web search problems in the future?

Not Useful				Very Useful
○	○	○	○	○
1	2	3	4	5

(6b) Briefly explain your reasoning for the rating you gave in the previous question.

(7) Rate the overall quality of the Strategy Scratchpad:

Poor				Excellent
○	○	○	○	○
1	2	3	4	5

Figure 3.3: Validation task for training Question Y with a real worker solution

3.4.1 Conditions

The five conditions we ran in the first experiment were as follows:

- **Control**: Workers receive no training. They are simply given instructions on how to perform the web search task and are then given the five test tasks (A, B, C, D, and E) in a random order.
- **Solution**: Workers are first presented with training tasks X and Y in a random order as a form of training. Workers are given the same instructions as in the control condition, except that it tells them they will have seven tasks instead of five. They are not told that the first two tasks are for training. (We refer to this as the *solution* condition as workers are *solving* additional tasks for training.)
- **Gold Standard**: Workers start by solving two tasks for training as in the solution condition. However, after submitting the answer to each of these two tasks, workers are shown the correct answer to the task along with an expert example solution, such as the one shown in Figure 3.2. Workers are told that the expert solutions are more thorough than what we expect from them.[3]
- **Example**: Workers are given two expert examples for training, which are the same as the expert solutions given in the gold standard condition. On the instructions given to workers for reviewing the examples, workers are informed that they cannot move on to the next task until 30 seconds elapse so that they are encouraged to spend time reading and understanding the examples. As in the gold standard condition, workers are also told that the examples will be more thorough than the task solutions we expect from them. Once they finish reading the examples, workers are given explicit instructions for completing web search tasks followed by the five test tasks.
- **Validation**: Workers are first asked to validate two other workers' solutions for questions X and Y in a random order. The solutions to be validated are randomly chosen from a pool of 28 solutions collected in a previous pilot study. In each validation task, a worker sees the answer, strategy scratchpad, and failed attempts box of the solution they are validating, and are then asked a series of questions about the solution to be validated, as shown in Figure 3.3. Once they complete the two validation tasks, workers are given explicit instructions for completing web search tasks followed by the five test tasks.

We paid workers between $0.50 and $1.50 for completing a web search task (depending on whether or not they got the correct answer and the completeness of their strategy), $0.50 for each validation task, and $0.10 for reviewing an expert example. Workers in the gold standard condition were only paid for solving the tasks and were not paid extra for reviewing examples, because we do not enforce them to read through the examples. Additionally, we paid workers $0.50 for

[1] We recruited only workers from the United States who were at least 18 years old and had at least a 98% approval rate.

[2] Every worker did only one HIT, which was composed of a series of tasks.

[3] Note that we do not refer to these tasks as gold standard tasks to workers since the term "gold standard" may have negative associations for workers in terms of disqualification or rejection of work.

Number of Workers (Percent of Workers that Start HIT)				
	Start HIT	Finish ≥ 1 training task	Finish ≥ 1 test task	Finish all tasks
Control	397	-	210 (0.53)	150 (0.38)
Solution	372	146 (0.39)	93 (0.25)	71 (0.19)
Gold Standard	372	142 (0.38)	95 (0.26)	72 (0.19)
Example	362	280 (0.77)	188 (0.52)	140 (0.39)
Validation	369	225 (0.61)	162 (0.44)	107 (0.29)

Table 3.1: Number of workers starting each condition in Experiment I and the retention rate at various points in the HIT

completing the survey. Workers who did not submit the survey were not paid at all, since their data could not be submitted to Mechanical Turk, which we made clear to workers.

3.4.2 Results

Table 3.1 shows how many workers were in each condition (i.e. how many went beyond the informed consent form) and the retention rates per condition: what percentage of workers did at least one training task, did at least one test task, and did all of the tasks. We see that the control and example conditions had the highest retention rates at all points in the HIT, and the solution and gold standard conditions had the least, with the validation condition in between. This is not surprising as the control condition has no training and the example condition offers the fastest form of training whereas the gold standard and solution conditions spend the longest time in the training phases. Workers may be more likely to drop out the longer they are in the task, and this could be due to either external factors that have nothing to do with the task or due to a variety of task-related factors such as boredom, annoyance with the task, the difficulty of the task, and/or the time spent appearing to be not worth the pay. All of these were expressed as reasons for dropping out in our survey. Nonetheless we find that even in the most time-consuming conditions (which took near an hour on average, but took up to two hours for some workers), nearly 20% of workers completed all tasks. Moreover, we find that in all conditions (except the control) around half of the workers who did at least one training task finished all of the tasks, suggesting that among workers who are willing to finish the first training task, there is roughly an equal proportion of highly committed workers in every condition.

Table 3.2 reports non-retention metrics for the various conditions. We are particularly interested in whether training increases the accuracy of workers on the test tasks, and if so, which forms of training are most effective at increasing worker accuracy. We report both the average per task accuracy (averaged over all test tasks) and the average accuracy per worker (among workers who did all five test questions). The average accuracy per worker is computed by first calculating the average accuracy for each worker on the five test questions they did, and then averaging this

	Per Test Task			Per Worker	
	Accuracy	Time (min)	Strategy Length (char)	Accuracy	Total Time (min)
Control	0.48	8.28 ± 7.35	492 ± 385	0.50 ± 0.27	41.2 ± 22.2
Solution	0.54	6.65 ± 6.33	477 ± 396	0.55 ± 0.28	55.2 ± 23.9
Gold Standard	0.51	6.69 ± 4.47	467 ± 297	0.52 ± 0.21	54.7 ± 20.8
Example	0.61	9.58 ± 7.15	625 ± 424	0.61 ± 0.26	49.6 ± 22.0
Validation	0.55	9.47 ± 7.32	539 ± 339	0.56 ± 0.26	57.3 ± 24.6

Table 3.2: Comparison across conditions in Experiment I on metrics of interest. Mean ± standard deviation is shown. Per task accuracy is a Bernoulli random variable; as accuracies are near 0.5, standard deviation is nearly 0.5 for every condition. Per worker columns only include workers who do all five test tasks, except for the training cost column, which is averaged over all workers who do both training tasks. The training cost column shows how much we paid workers for training on average. Note that workers in the example and validation conditions were paid a fixed amount.

	Question A	Question B	Question C	Question D	Question E
Control	0.67	0.43	0.50	0.53	0.29
Solution	0.70	0.49	0.57	0.62	0.35
Gold Standard	**0.84**	0.26	0.62	0.59	0.25
Example	0.77	**0.50**	**0.72**	**0.65**	**0.42**
Validation	0.73	**0.50**	0.54	0.64	0.34

Table 3.3: Comparison across conditions in Experiment I of per task accuracy for each question. The condition with the highest accuracy for each question is bolded.

measure across the workers.[4]

We find that for both measures of worker accuracy, all training conditions outperformed the control condition of having no training. The differences in per worker accuracy were significant based on the non-parametric Kruskal-Wallis test ($p = 0.0067 < 0.05$). Doing a post hoc analysis on the per worker accuracy using Mann-Whitney U tests, we find that the example condition was significantly better than the control after a Bonferroni correction for doing four tests. With a similar analysis on per task accuracy using two-proportion z-tests[5], we find that the example and validation conditions were significantly better than the control after a Bonferroni correction.

The example condition had the highest gains in accuracy over the control condition with an effect

[4] The accuracy per worker for workers who did *at least one task* yields similar results. However, it is a more noisy measure since workers who did only one task have a much more noisy accuracy than workers who did all five, but in the aggregate average across workers, accuracy rates for workers who completed 5 tasks would be weighted equally with those that completed 1 task.

[5] Not all of the assumptions of this statistical test are satisfied in our domain as answers for the same worker on different questions are dependent.

size of 0.25 (Cohen's *h*) for per task accuracy, which is considered a small effect, and 0.42 (Glass' Δ) for per worker accuracy, which is closer to a medium effect. While these effect sizes are not considered large in the educational literature, we note that our form of training is *much* shorter than traditional educational interventions, so we do not expect effect sizes to compare to those of traditional interventions.

As for time spent per test task, we find that the example and validation conditions took longer than the control by over a minute on average, while the solution and gold standard conditions took less time than the control by over 1.5 minutes on average. Despite the large difference in time per task, we find that in total, the example condition took less time on average for workers who did all of the tasks than the solution and gold standard conditions since the example condition spends much less time on training. Furthermore, the number of characters in the strategy scratchpad was greater for the example and validation conditions than the other conditions.

Finally, we do a comparison of the conditions on the per task accuracy for each of the five test questions, as reported in Table 3.3. We find that the example condition achieved the highest per task accuracy on all questions except for Question A, where the gold standard condition did much better than any other condition. On the other hand, we find that the gold standard condition did much poorer on Question B compared to all the other conditions. In the discussion section, we present a case study analyzing why the effectiveness of the gold standard condition may vary between tasks.

3.5 Experiment II

The results of Experiment I demonstrating the effectiveness of the example and validation conditions suggest that there might be hope for the validation condition to perform as well as the example condition if we only present workers with the "best solutions" to validate. This experiment will show how validating peer solutions can possibly be as effective or even more effective than reading expert examples, and will provide preliminary evidence for the potential impact of content curation that we will explore more fully in the next chapter.

3.5.1 Filtering Validation Tasks

We seek to answer the question "what properties of a solution makes it beneficial for training when presented as a validation task?" To help answer this question, we performed linear regression on a set of features for each of the solutions that was validated in Experiment I[6] to see how well they predict the per task accuracy of workers who validated that particular solution. The features for each validated solution include the answers provided for each quantifiable question

[6] We removed one one of the solutions that was a clear outlier. It had the longest solution, but the workers who validated it had a lower average accuracy than workers who validated any other solution, which violates the trend we discuss below. In addition to being a bad solution, it was formatted very strangely (without newline characters) and its length was due to long URLs; this seems to have had a negative effect on workers.

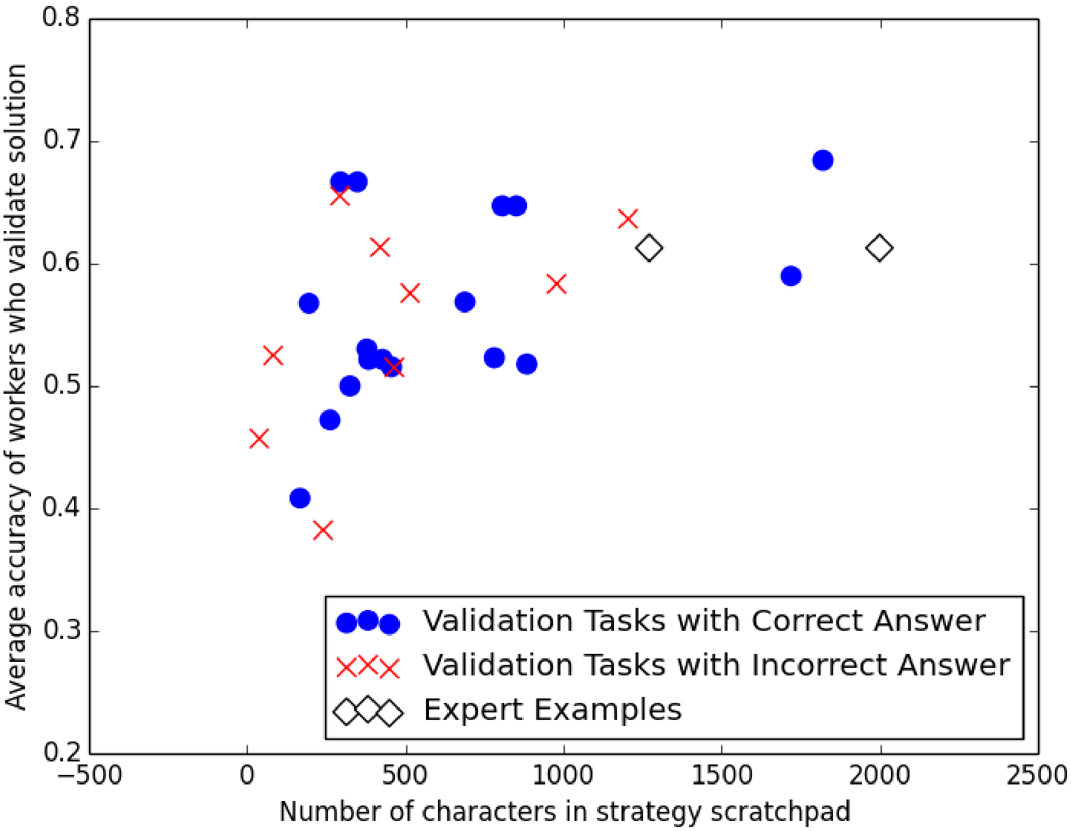


Figure 3.4: Average per worker accuracy on tasks done after seeing a particular validation task for training vs. the number of characters in the strategy scratchpad for that validation task. Each point represents a particular solution given as a validation task. The blue circles show solutions that arrived at the correct answer and the red x's show solutions that arrived at the wrong answer. The diamonds indicate the two expert solutions provided in the example condition for comparison; the average accuracy in this case is for all workers in the example condition.

asked in the validation task (see Figure 3.3) averaged over workers who validated that solution. To this set of features we also added the number of characters in the strategy scratchpad for that task, the number of characters in the failed attempts box for that task, and the amount of time the worker who authored the solution spent solving that task. We performed regularized linear regression (LASSO with a regularization parameter that was chosen using Leave-One-Out cross-validation). The resulting analysis indicated that only the number of characters in the strategy scratchpad was correlated with accuracy[7].

Figure 3.4 shows for each solution presented as a validation task, the per worker accuracy (in the testing phase) of workers who validated that solution vs. the number of characters in the strategy scratchpad for that solution. The Pearson correlation coefficient is 0.46. We also see

[7] That is, the LASSO assigned a coefficient of 0 to all other predictors.

from the plot that whether the solution had a correct or incorrect answer does not seem clearly correlated with the later accuracy of workers who validated it. This suggests that in this setting, regardless of solution correctness, longer solutions are generally more effective for training. Thus a requester could potentially decide whether a solution should be given for training as soon as the solution is generated, by checking how long it is, without needing to first assess if the solution is correct.

Since our goal was to mimic the training process followed in Experiment I, in which all training conditions involved two tasks, our next task was devising a method for automatically identifying good *pairs* of validation tasks to present workers. We split the solutions into "short" and "long" ones by whether the solution length was longer or shorter than a single handset threshold. When we analyzed the effect of the different orderings of short and long solutions on worker accuracy on the data collected from Experiment I, we found that presenting a short solution followed by a long solution appears better than the other combinations for various thresholds. We note that we had very little data to evaluate presenting two long solutions, so it may have actually been the best option, but we chose the more conservative option that was supported by our data. Choosing to present a short solution followed by a long one also has the practical advantage that all solutions collected from prior workers can be validated, resulting in automated quality control for all solutions collected from crowdworkers. In our second experiment, we test the efficacy of this approach for filtering solutions that we present workers.

3.5.2 Experimental Design

Experiment II compared three conditions: **example-II**, **validation-II**, and **filtered validation**. Example-II and validation-II are the same as the corresponding conditions from the first experiment with a new worker pool. To see how the trends from Experiment I generalize when a new set of solutions is provided for validation, we refreshed the solution set for validation-II with solutions collected from Experiment I. The set included 100 solutions to Questions X and Y randomly sampled from those collected from the solution condition of Experiment I as well as the 28 solutions used in the validation condition of the previous study.

The solutions used in the filtered validation condition came from the same randomly sampled set of 100 solutions generated in Experiment I. As before, the ordering of questions X and Y was randomized. The first solution each worker validated was chosen from among those that had fewer than 800 characters, and the second solution they validated was chosen from among those that had at least 800 characters. This threshold of 800 characters resulted in 76 short and 24 long solutions used in the filtered validation condition.

3.5.3 Results

Table 3.4 displays how many workers were in each condition and the retention rates in each condition. Although our main focus is on how conditions compared within Experiment II, we note that the example-II condition had a lower retention rate than the earlier example condition,

Number of Workers (Percent of Workers that Start HIT)				
	Start HIT	Finish ≥ 1 training task	Finish ≥ 1 test task	Finish all tasks
Example-II	310	239 (0.77)	150 (0.48)	102 (0.33)
Validation-II	330	189 (0.57)	140 (0.42)	95 (0.29)
Filtered Validation	314	195 (0.62)	142 (0.45)	88 (0.28)

Table 3.4: Number of workers starting each condition in Experiment II and the retention rate at various points of the HIT

	Per Test Task			Per Worker	
	Accuracy	Time (min)	Strategy Length (char)	Accuracy	Total Time (min)
Example-II	0.59	8.66 ± 7.25	550 ± 379	0.59 ± 0.26	42.6 ± 20.0
Validation-II	0.57	9.02 ± 6.81	561 ± 362	0.58 ± 0.23	53.5 ± 22.1
Filtered Validation	0.59	9.58 ± 7.87	618 ± 415	0.60 ± 0.25	52.4 ± 21.5
Filtered Medium-Long	0.69	10.96 ± 10.50	692 ± 424	0.74 ± 0.17	55.4 ± 21.6

Table 3.5: Comparison across conditions in Experiment II on metrics of interest. Mean ± standard deviation is shown.

indicating the worker pool may have slightly changed. The validation-II and filtered validation conditions have similar retention rates.

Table 3.5 presents non-retention metrics. The example-II and filtered validation conditions had nearly identical performance on per task and per worker accuracy. These conditions perform slightly better than the validation-II condition, but the differences are not significant. Interestingly, there may be a regression to the mean effect between the first and second experiment, as the difference between the standard validation and example conditions in Experiment I was larger (0.06 for worker accuracy) than the difference between validation-II and example-II (0.02).

In Experiment I, we had a limited number of longer task length solutions provided to workers to validate, thereby limiting our ability to explore the effects of providing workers with two longer tasks to validate. However, a number of the solutions presented to workers in Experiment II (i.e. solutions generated during Experiment I) had a longer length, and so we can now analyze how well workers who were provided with only medium and long solutions performed. To do so, we selected the subset of workers in the filtered validation condition whose first task was to validate a solution between 500 and 800 characters long (since the first task was never longer than 800 characters by design), and whose second task was to validate a solution that was at least 1000 characters long (n=34 workers). We refer to this subset of workers as the **filtered medium-long** group.

We find that workers in the filtered medium-long group have a much higher average per task accuracy (0.69) than the example-II condition (0.59), validation-II condition (0.57), and filtered validation condition (0.59). The difference is significant ($p < 0.05$) between the filtered medium-long group and validation-II condition after doing a Bonferroni correction for multiple tests. The effect size of per task accuracy for the filtered medium-long workers as compared to the example-II condition was 0.19 (Cohen's h) and the effect size for per worker accuracy between the two conditions was 0.55 (Glass' Δ). The average time per test task and average strategy length were also considerably larger for these workers than for workers in all three of the actual conditions.

3.6 Discussion and Conclusion

We compared the efficacy of various forms of training for complex problem solving tasks. In our first experiment, we found that using expert examples was the most effective form of training as captured by several metrics, including increasing the accuracy of workers on the task. We then showed that having workers validate crowdsourced solutions that are beyond a threshold length can be even more effective than having them read expert examples.

Follow-up studies on training may be informative to better understand the nature of cognitive processes involved in training for complex tasks. For example, it is not clear to what extent the validation process is essential to the training benefits of the validation task. Perhaps we could simply present long crowdsourced solutions to workers as expert examples. However, we hypothesize that the validation process is useful, in part because it provides workers a "rubric" of what constitutes a good solution. This was also seen in the work of Dow et al. [2012] where workers who self-assessed their work or had an expert assess their work had similar performance gains, possibly because both groups saw similar rubrics. We also expect that the validation task encourages actively engaging with an example rather than passively reading it, which could be more beneficial for learning [Chi and Wylie, 2014]. Validating peer work requires more cognitive load than reading an expert example, but likely requires less cognitive load than solving web search tasks from scratch. Therefore, based on the expertise-reversal effect [Kalyuga et al., 2001], perhaps validating peer work is most effective for workers who have some familiarity with web search, but would struggle to complete a web search task on their own, which we imagine is the case for many crowdworkers.

While our focus has been on learning from peer solutions, workers can likely also learn from their own work. As such, we expect that worker performance may have improved simply because we asked workers to document their strategies, which can be viewed as a kind of self-explanation [Chi et al., 1989]. Would our results hold if we no longer have workers document their strategies after training?

Finally, we think the most practically important future direction is to run similar experiments across a series of domains to see if our results generalize. In particular, it would be interesting to find if filtering by solution length is effective in all domains, and if not, if we can find a general machine learning protocol for finding the features of high-quality validation tasks in any domain. We hypothesize that this is possible, and if so, that we can create crowdsourcing platforms that automatically learn to train unskilled workers. As one step in this direction, in the next chapter, we examine the efficacy of filtering high quality solutions based on crowdworker ratings in a different complex crowdsourcing task.

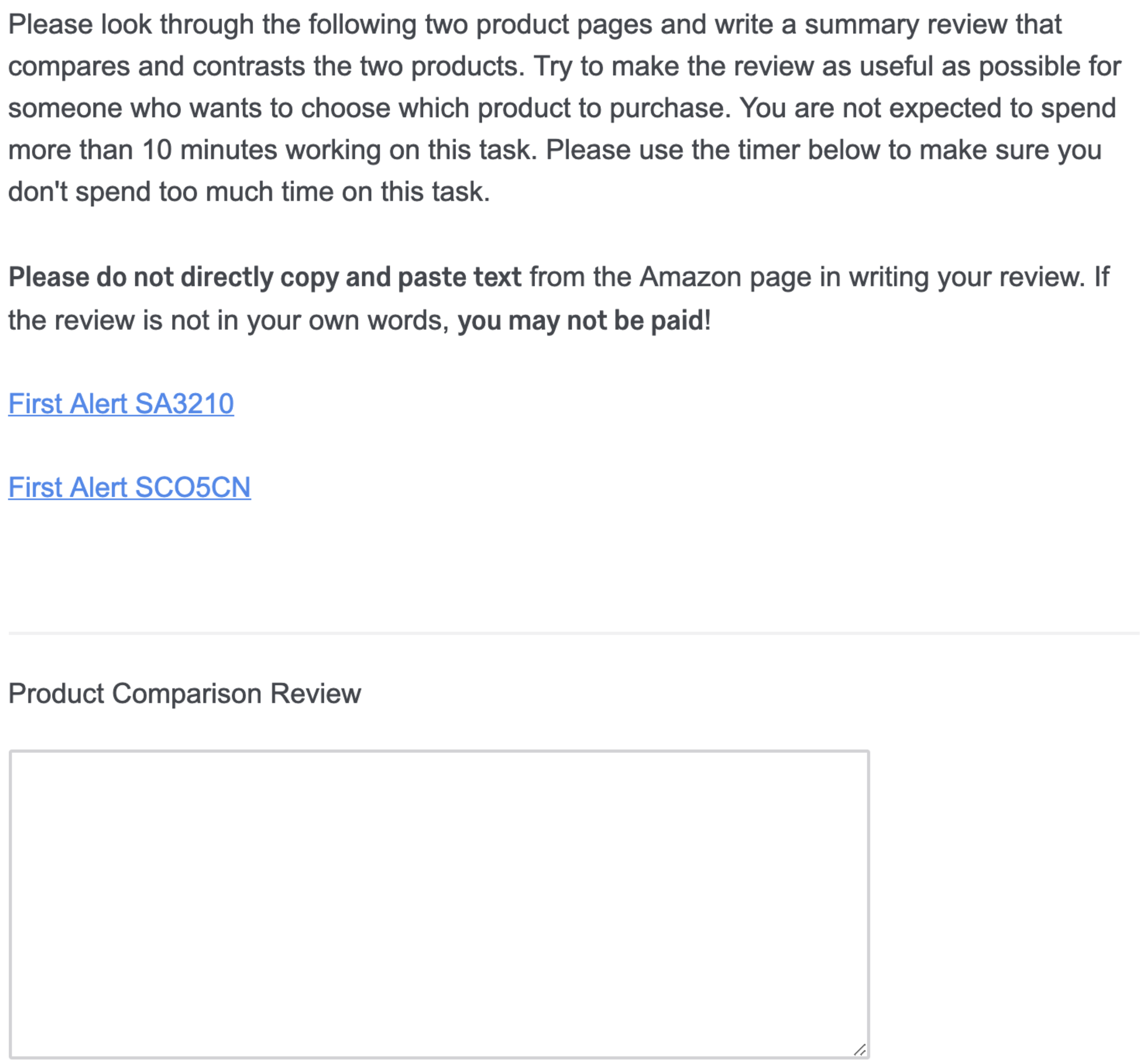
Chapter 4

Reading Peer Examples[*]

> The world is often unkind to new talent, new creations. The new needs friends. Last night, I experienced something new, an extra-ordinary meal from a singularly unexpected source. To say that both the meal and its maker have challenged my preconceptions about fine cooking is a gross understatement. They have rocked me to my core. In the past, I have made no secret of my disdain for Chef Gusteau's famous motto: 'Anyone can cook.' But I realize, only now do I truly understand what he meant. Not everyone can become a great artist, but a great artist can come from anywhere.
>
> Anton Ego, *Ratatouille*

While the previous chapter was focused on complex problem solving tasks, here we turn to more creative, open-ended writing tasks. In particular, we examine how to effectively use peer-generated examples in the context of a task where crowd workers read two Amazon product pages and write a review that compares and contrasts the two products. Since these tasks are more open-ended, the work cannot be validated in the same way as in complex problem solving tasks, where a worker can follow another worker's solution steps to see if they arrived at the correct answer. Instead, we wanted to see if simply presenting worker product reviews as peer-generated examples is an effective form of training in this context. Our initial aim was to compare the efficacy of various ways of presenting peer-generated artifacts (i.e., showing a single example, showing pairs of examples, and showing worker-generated guidelines) to prepare workers for *transfer tasks* (i.e., writing product comparison reviews for very different kinds of products). However, we found in our first experiment that randomly presenting peer-generated examples or guidelines was not very effective, presumably because the average quality of peer-generated examples was low. We ran a second experiment that showed that peer-generated examples can be useful when examples that are of sufficiently high quality are shown, at least on a *near transfer* task (i.e., a task that closely resembles the example).

[*] The work described here was done in collaboration with Ece Kamar and Emma Brunskill.



Please look through the following two product pages and write a summary review that compares and contrasts the two products. Try to make the review as useful as possible for someone who wants to choose which product to purchase. You are not expected to spend more than 10 minutes working on this task. Please use the timer below to make sure you don't spend too much time on this task.

Please do not directly copy and paste text from the Amazon page in writing your review. If the review is not in your own words, **you may not be paid**!

First Alert SA3210

First Alert SCO5CN

Product Comparison Review

Figure 4.1: Product comparison task. The task here is to compare smoke alarm products.

Further, our analysis seems to indicate that even among high quality examples, there are differences in how pedagogically effective they seem to be. Our preliminary analysis seems to suggest that surface-level features like the format of the example may cause some examples to be less effective than others, but more work is needed to confirm this. This suggests the need to automatically search the space of peer-generated examples to find ones that are more effective, which we motivated in the previous chapter. By automatically discovering what makes a learnersourced example pedagogically effective, future work could not only serve the practical goal of bootstrapped example creation, but also serve the scientific goal of determining what makes a great example great.

4.1 Task Design

The domain in this task is writing open-ended product comparison reviews. Workers are given links to two Amazon pages for products that are somewhat similar (but with several salient differences). Writing product reviews is a crowdsourcing task that has been used in prior research in training crowd workers [Dow et al., 2012, Steuer et al., 2017, Zhu et al., 2014]. Prior work had crowd workers review products that they own, but this limits the ability to use crowdsourcing to review particular products. Therefore, we wanted workers to review specific products available on Amazon. We chose to have workers compare two products, instead of reviewing one, both because the comparative nature of the task would help ground the content of the review and because product comparisons are a common service that many websites provide.

Figure 4.1 shows an example of a product comparison task with instructions as was presented to workers. As can be seen from the instructions, that task was intentionally left open-ended. This was for several reasons. First, we wanted workers to have some creativity in determining how to write the review; a good review can come in many shapes and forms, and the worker should be able to determine that. Second, the particulars of a good review might vary when comparing different types of products, and so we do not want to give overly specific instructions that might limit the worker's ability to write a good review for a new product type. Relatedly, the requester might not know what a good review would look like for all types of products. However, the task is not meant to be completely subjective; we do give workers a metric for assessing quality of a review—usefulness for potential buyers who want to choose which product to purchase. As such, to evaluate the quality of reviews, we had workers grade the reviews on a scale that tries to assess how useful a review would be for potential buyers.

We had workers write reviews for three distinct categories of products: smoke alarms, board games, and gluten-free macaroni and cheese products. We choose product categories that are very different from one another so that we could test both whether seeing examples of reviews for products from the same category helps as well as whether seeing examples from products of a different category is still useful. We were particularly interested in identifying how to best support transfer to other categories, as one can imagine crowdsourcing requesters may need to have workers complete many similar tasks but for different categories, and the categories of interest might change over time. We used five tasks in particular: two for smoke alarms (Tasks A and A'), two for board games (Tasks B and B') and two for mac and cheese (Task C). Tasks A and B were used to collect an initial pool of examples that we could show workers, and Tasks A', B', and C were used to test workers to see if examples improve their performance.

4.2 Hypotheses

Our first experiment was guided by three main hypotheses that were supported by the psychology literature.

Hypothesis 1 *Seeing peer-generated artifacts leads to improved performance on future tasks.*

Motivated by results from Spiro and colleagues, we additionally hypothesize that it is important to see multiple peer-generated examples for far transfer tasks in ill-structured domains [Spiro and DeSchryver, 2009, Spiro et al., 1988].

Hypothesis 2 *Seeing multiple peer-generated examples from different categories will lead to greater performance than seeing a single peer-generated example on far transfer tasks (i.e., tasks that are from different categories than the single example).*

Additionally, prior work has shown that giving guidelines that are abstracted from solving different tasks will lead to greater transfer than examples, but is less beneficial for performance support and near transfer [Eiriksdottir and Catrambone, 2011], thus motivating our final hypothesis.

Hypothesis 3 *Seeing peer-generated guidelines will lead to greater performance on far transfer tasks than seeing a single peer-generated example, but will have worse performance on near transfer tasks.*

While these hypotheses guided the design of the first experiment, I find that none of these hypotheses were met, perhaps due to the low quality of random peer-generated examples, which was not an issue in prior work on ill-structured domains, since prior work used expert worked examples. Thus, the focus of this chapter will be more on how the *quality* of examples affects performance on future tasks, rather than how to best support transfer. However, I will also discuss what our results seem to suggest in terms of best supporting transfer.

4.3 Experimental Design

All of our experiments were conducted on Amazon Mechanical Turk. Workers who accepted our Mechanical Turk Human Intelligence Task (HIT) were first given a consent form and randomly assigned to one of several conditions (depending on the experiment). After giving consent, workers were given instructions followed by some form of training (seeing examples or guidelines) depending on the condition they were assigned to (unless they were in the control condition). Workers were asked to spend as much time as they needed reading the examples or guidelines they were presented. They then do up to three tasks in order (A', B', and C) followed by a short survey with qualitative questions about the difficulty of the task and usefulness of training. Workers were free to stop working on the tasks at any point in time, at which point they were given the survey. Workers were paid $0.50 just for doing the HIT and survey, in addition to $2 for each product comparison review they completed. They were told that they would receive the bonus payment provided that they follow the instructions. Additionally, workers were given $0.50 for each example or guidelines that they had to read. The pay was chosen so that workers could expect a wage of $11/hour[1] if they spent 10 minutes on each review, and as such workers were suggested to not spend more than 10 minutes writing each review.

[1] This is in line with Dynamo's ethical standards for paying crowdworkers: `http://wiki.wearedynamo.org/index.php/Fair_payment`.

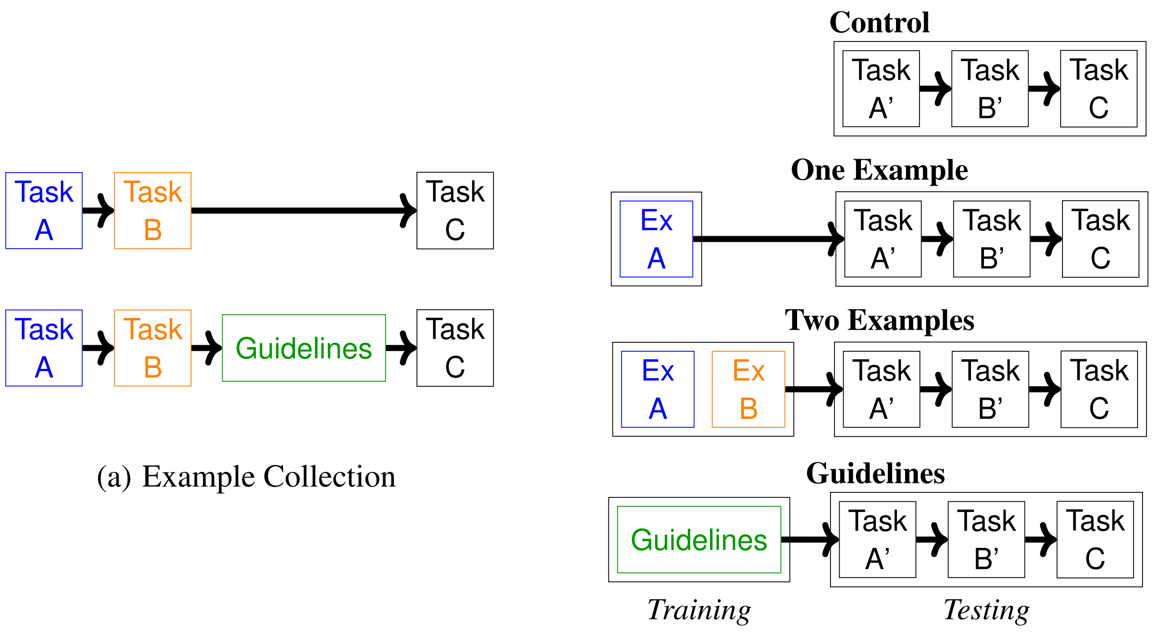
After each experiment, we had workers grade the solutions both in terms of overall quality as well as checking whether the review contained specific features. We first released a HIT to test workers' abilities to accurately grade some gold standard product comparison reviews. Only workers who had participated in the associated experiment were allowed to take the HIT. Workers who were qualified were then given access to HITs for each review that needed to be graded. Each review was graded by three to five different workers. The rubric consisted of two parts. First, the graders were asked whether or not the review mentioned particular points that we believed were either *generally* good to mention in a product review (i.e., price, average star rating, and number of reviews on Amazon) or important to mention when comparing those *particular* products (e.g., mentioning that both mac and cheese products are gluten free or mentioning that one of the products is vegan/dairy-free but the other is not). Second, the graders were asked to rate the overall quality of the solution using the following scale:

5. It's hard to imagine a more useful resource for someone to decide which product to buy. The review appears to contain no factual errors.
4. The review would help you decide which product is best, but could have had some more information or could have been structured better.
3. The review would be helpful, but you would need to do more research to decide which product to buy.
2. The review has some distinctions between the two products, but you basically need to do your own research from scratch to decide which product to buy.
1. The review is misleading or does not really contain useful information (e.g., contains a major factual error that could result in purchasing the wrong product).

This overall quality scale is our primary metric for evaluating the efficacy of different types of training.

4.4 Example Collection

We recruited 70 Mechanical Turk workers to complete up to three tasks (A, B, and C). In addition, to collecting examples from tasks, we also wanted to collected general guidelines for doing product comparison review tasks which we could also test. Workers were randomly assigned to one of two conditions. In one condition, workers just completed the three tasks, but in the second condition, after doing two tasks, workers were asked to write down general guidelines for doing the tasks. The experimental design is shown in Figure 4.2(a). We randomly assigned the workers to these two conditions in order to assess whether writing guidelines is beneficial to the workers who write the guidelines themselves, namely whether it improves their performance on Task C. 56 workers submitted acceptable work; work was rejected when workers clearly did not put an honest effort into the task for example by copy-pasting text from the Amazon product pages (which the instructions explicitly told them not to do). This process resulted in 56



(a) Example Collection

(b) Experiment 1

Figure 4.2: Experimental design for (a) the example collection phase and (b) Experiment 1. The reviews and guidelines generated from the example collection phase are the examples and guidelines used in Experiment 1.

reviews for Task A, 47 reviews for Task B, and 41 reviews for Task C (which were not used as examples).

Each review was graded by five workers. To measure the inter-rater reliability, we computed the intraclass correlation coefficient, namely ICC(1, k), which measures how likely it is that two randomly chosen samples of k graders would assign the same quality (1-5) for a given review [Shrout and Fleiss, 1979]. The intraclass correlation for the example collection phase as well as the subsequent experiments is shown in Table 4.1. In the next two experiments, we only used 3 workers to grade each review, which could explain why the correlation was a little lower. However, overall the intraclass correlation tends to be high, meaning the average overall quality per review can be a reliable way of measuring review quality.

There appeared to be no difference between the two conditions in the overall quality of Task C (2.4 for workers who created guidelines vs. 2.42 for workers in the control condition).

4.5 Experiment 1: Random Examples

In our first experiment, we wanted to test whether randomly presenting examples or guidelines improves the performance of workers on future tasks. In addition, we wanted to use this experiment to test our hypotheses on what types of peer-generated artifacts are effective for training,

	Task A	Task B	Task C
Example Collection	0.86	0.86	0.90
	Task A’	Task B’	Task C
Experiment 1	0.74	0.75	0.67
Experiment 2	0.75	0.87	0.88

Table 4.1: Intraclass correlation coefficients, ICC(1, *k*), of overall quality ratings given to reviews from each experiment

Condition	Num of Workers	Mean Overall Quality ± Std Dev		
		Task A’	Task B’	Task C
Control	92	2.43 ± 0.7	3.00 ± 0.7	*2.58 ± 0.7*
One Example	102	*2.27 ± 0.7*	*2.85 ± 0.7*	2.66 ± 0.7
Two Examples	97	**2.47 ± 0.7**	**3.02 ± 0.7**	2.67 ± 0.7
Guidelines	101	**2.47 ± 0.6**	2.94 ± 0.7	**2.68 ± 0.7**
One Example ≥	52	2.40 ± 0.7	2.87 ± 0.7	2.68 ± 0.8
One Example <	50	2.14 ± 0.7	2.83 ± 0.6	2.64 ± 0.7
Two Examples ≥, ≥	23	2.57 ± 0.8	**3.09 ± 0.8**	**2.86 ± 0.7**
Two Examples ≥, <	24	**2.65 ± 0.5**	3.01 ± 0.7	2.74 ± 0.7
Two Examples <, ≥	27	2.47 ± 0.7	3.07 ± 0.6	2.45 ± 0.6
Two Examples <, <	23	2.20 ± 0.6	2.90 ± 0.7	2.61 ± 0.8

Table 4.2: Experiment 1 Results. The highest-performing condition for each task is shown in bold and the lowest-performing condition is italicized. The highest-performing median-split condition for each task is also shown in bold. ≥ indicates above-median (greater than or equal to median) example quality and < indicates below-median example quality.

both for near-transfer tasks as well as for transfer tasks. As such, we randomly assigned workers to one of four conditions. In the **control** condition workers received no training, in the **one example** condition, workers saw a single randomly chosen example from Task A, in the **two examples** condition, workers saw a randomly chosen example from Task A followed by a randomly chosen example from Task B (both presented on the same page), and in the **guidelines** condition, workers saw randomly chosen guidelines generated by workers (after completing Tasks A and B). These conditions are depicted in Figure 4.2(b). The workers were told that the examples and guidelines they saw were randomly chosen, so that workers would not be confused if they saw bad product comparison reviews. For examples, workers in the one example condition were told “The product comparison review shown below was randomly selected from ones submitted by other workers, so it is not necessarily of high quality. Good and bad examples can both help inform your work. If the peer example is bad, think about ways it could be improved.” In this way, we were also hoping that seeing bad examples could sometimes be instructive, which would potentially increase the value of randomly presenting peer-generated examples. Workers were also given the average overall quality score for each example they saw.

4.5.1 Results

We recruited 433 participants from Mechanical Turk, of which 416 submitted acceptable work and 392 completed at least one of the test tasks. The results are shown in Table 4.2. Since, workers were allowed to stop working on the tasks at any time, the number of workers decreases after each task. Among the workers who did the first task, 86% of workers do all three tasks in the control and two examples conditions, 75% in the one example condition, and 73% in the guidelines condition. For each task, we ran a Kruskal-Wallis test comparing the four conditions, and found that there is no significant effect of condition for each task ($p = 0.09$ for Task A' with larger p-values for the other two tasks). However, we observe that seeing a single example appears to be no better than control, and trends worse on Tasks A' and Task B'. Seeing two examples or guidelines generally results in performance that is comparable to or better than not receiving training.

We hypothesize that the reason none of these conditions appears effective and that a single example might even be bad is that these examples and guidelines were randomly chosen. To analyze this further, we looked at the results depending on whether the examples shown to workers were of above median or below median overall quality. The median example quality for Task A was 2.05 in the one example condition and 2.1 in the two examples condition and the median quality for Task B was 3 in the two examples condition. The results for these median-split conditions is shown at the bottom of Table 4.2. We observe the following trends:

- Seeing an example of above median quality appears better than seeing an example of below median quality on average. This is clearest when comparing two examples that are both above median quality to two examples that are both below median quality.

- Seeing a single example does worse than the control on Task A' and B' regardless of whether the example was of above or below median quality.

- Seeing two examples of above median quality has the highest or second highest performance of all above/below median splits.

- For Task A', seeing an above median example on Task A is better than below median, regardless of the quality of the example on Task B, and analogously for Task B', seeing an above median example on Task B is better than below median, regardless of the quality of the example on Task A.

Taken together, these trends seem to suggest that seeing random examples may not help (or only marginally help), but seeing two high quality (e.g., above median) examples is likely to lead to more learning gains. Moreover, the results suggest that seeing a good example from the same category as the task to be completed is helpful (e.g., seeing a good example from Task A is helpful in doing Task A'). Given that these are just trends, we run a second experiment to test if seeing two high quality examples is actually beneficial.

4.6 Experiment 2: High Quality Examples

To test if only presenting high quality reviews is pedagogically effective, we decided to try the simplest condition that we had reason to believe would have the highest performance benefits over not receiving any training. Thus, we chose to show two examples (one from Task A and one from Task B as before) that were both among the highest quality for their respective task. In particular, for each task, we randomly chose one of the three examples that had the highest overall quality score. We did not want to simply use the best example, because it would be possible that that particular example would have been ineffective for some reason, and hence the experiment would have been uninformative. On the other hand, we were also hoping to discern if the exact examples that were seen would make a difference, and so we wanted to ensure we would have several workers see each example. Using three examples for each task seemed to balance these two competing goals. For Task A, the three best examples had average quality scores of 3.9, 3.9, and 3.6, and for Task B, the three best examples had average overall quality scores of 4, 3.8, and 3.8[2] Therefore, in terms of overall quality, these examples were all comparable and much higher than the median overall quality for each task (2.2 for Task A and 2.8 for Task B). The three examples for each task are shown in Figures 4.3 and 4.4.

4.6.1 Results

We ran an experiment comparing the new **two good examples** condition with a control condition as before. Other than the examples shown, everything was the same as the previous experiment. We recruited 161 workers on Mechanical Turk, of which 151 completed at least one task. The results are shown in Table 4.3. We note that there seems to be a decrease in performance compared to Experiment 1 (which can be seen by comparing the two control conditions). This could either be due to a change in the worker pool or due to biases in grading. We note that Experiment 1 had lower inter-rater reliability, so perhaps the grading was deflated in that experiment. However, given the reliability is relatively high, we do not think this largely impacts the results. As before, the number of workers drops after each task: 84% of workers finished all three tasks in the control condition, and 71% of workers complete all three tasks in the two good examples condition. While the completion rate for the control is comparable to the previous experiment, the completion rate of the two good examples condition is considerably lower than that of the two examples condition in Experiment 1. This could be due to some workers dropping out due to high quality examples setting a high bar for the work they need to do, however, we do not know for certain if that is the case.

[2] There were other examples for Task A with a score of 3.6 and for Task B with a score of 3.8, so we just chose some particular examples that we thought might lead to interesting results.

If you’re looking for a hardwired solution, The First Alert is designed for that, plus it has a battery backup, in case you lose power. The First Alert is also the cheapest option.

The COOWOO Smoke Alarm is battery only, and while the manufacturer claims the battery will last 10 years, I seriously doubt that, and in general, you should replace your smoke alarm before then, anyways.

The First Alert, however, uses an ionization sensor, which tends to go off more frequently from cooking or other sources, and isn’t recommended for more modern setups, while the COOWOO is photoelectric, which is considered more reliable, and has less of a chance of going off from someone burning their food on the stove.

I recommend the COOWOO for that purpose, despite it being slightly more expensive.

(a) Example A1 (3.9)

First of all, the price difference:
First Alert BRK 9120B Hardwired Smoke Alarm with Battery Backup - $12.56
10 Years Battery-Operated Smoke and Fire Alarm/Detector(Not Hardwired) with Silence Button and 10-Hours Eliminates Late Night Low Battery Chirps Mode Photoelectric Sensor & UL Listed Smoke&Fire Alarm - $$19.99

First Alert has a backup battery, so it will work in case of a power outage. It also has an ionization sensor, which detect smoke reliably. It can connect with other compatible alarms so it will all sound when one detects smoke. 10 year limited warranty.

10 Years Battery-Operated Smoke and Fire Alarm/Detector is not hard wired, so it will be easier to install. It has a long battery life of 10 years. 7 year warranty. The design of the case is more sleek and modern looking than First Alert.

(b) Example A2 (3.6)

COOWOO Smoke Alarm versus First Alert Smoke Alarm. One big difference in these two smoke alarms is that the COOWOO is not hard wired. Instead, the CooWoo is operated by a 10-year lithium battery whereas the First Alert is hard wired and relies on a 9-volt battery back up to keep your family safe during a power outage. The first alert also has an ionization sensor which helps to detect fast flaming fires.

The First Alert can be interconnected with up to 12 other compatible smoke alarms and six compatible devices such as repeaters, door closers, bells, and horns. If one unit triggers an alarm, all the smoke alarms will sound. There is also an indicator which will show which unit initiated the alarm.

On the other hand, the COOWOO has a photoelectric Sensor and an alarm sensitivity of 1.0 2.52%/ft. OBS. When the smoke alarm device detects particles of combustion and the concentration of smoke reaches the alarm threshold, the red LED flashes once per second and emits a loud pulsating alarm until the smoke is cleared. It has an alarm volume of &gt;85dB(A) 3 meters.

(c) Example A3 (3.9)

Figure 4.3: Three examples for Task A (comparing First Alert and COOWOO smoke alarms) used in Experiment 2. The overall quality of each example as rated by workers is shown in parentheses.

Ticket to Ride tries to emulate cross country train journey in a board game while in Pandemic players work together to stop diseases from spreading. Both support similar number of players i.e. Pandemic 2-4 and Ticket to Ride 2-5. Play time is also similar with Pandemic advertising 45-60 and Ticket to Ride 30-60 minutes of game play. Both games are recommended for players aged 8 and above while the cost of Ticket to Ride is slightly more $49.99 than Pandemic $39.99. Main difference between the two games is obviously the game setting which is vastly different and hence game selection is mainly dependent on players preferences for that particular setting.

(a) Example B1 (3.8)

Pandemic and Ticket to Ride are both well reviewed board games worth considering. Pandemic is listed at $35.97 and is suitable for 2 - 4 players with a game running about 60 minutes. Ticket to Ride is priced at $44.97 and a game runs 30 - 60 minutes for 2 - 5 players.

Ticket to Ride is train adventure strategy game with the user traveling around the United States in a takeoff of "Around the World in 80 Days" with a winner takes all format. In Pandemic four diseases have broken out and players must work together as a team of specialists to save the world.

So, if you are in the mood to compete against other players, Ticket to Ride would be your choice. If you would prefer a cooperative game where all players win or lose together, Pandemic would be the better fit.

(b) Example B2 (3.8)

Today I am comparing two board games, Ticket to Ride and Pandemic, both which I personally own and love!

Both games will require at least 2 players to play! The nice thing about both of these games is that the game time is relatively fast and either game wont take more than 60 minutes to complete and both games are for ages 8+!

The main difference is that in Ticket to Ride you are playing alone, competing against everyone else to try and win, while in Pandemic you are working together to try and win the game! If you are trying to budget, Pandemic is also about $10 cheaper than Ticket to Ride.

Either way, you cannot go wrong as both games have over 4 stars and thousands of reviews!

(c) Example B3 (4)

Figure 4.4: Three examples for Task B (comparing the board games Ticket to Ride and Pandemic) used in Experiment 2. The overall quality of each example as rated by workers is shown in parentheses.

For each task, we ran a Mann-Whitney-U test to see if the two good examples condition had significantly higher median overall quality than the control. The result was significant for Task A’ (p = 0.005) with a Glass’ Δ effect size of 0.4, but not for the other two tasks. However, the trend seems to indicate that two good examples is better than control for the other tasks as well.

		Mean Overall Quality ± Std Dev		
Condition	Num of Workers	Task A’	Task B’	Task C
Control	82	*2.23 ± 0.8*	*2.62 ± 0.8*	*2.57 ± 0.9*
Two Good Examples	69	**2.53** ± **0.7**	**2.72** ± **0.9**	**2.76** ± **1.0**
Example A1	18	2.26 ± 0.7	2.40 ± 0.8	2.58 ± 0.9
Example A2	27	2.60 ± 0.7	**2.83** ± **1.0**	2.81 ± 1.1
Example A3	24	**2.64** ± **0.7**	2.81 ± 0.7	**2.83** ± **0.9**
Example B1	20	2.45 ± 0.8	2.35 ± 0.9	2.67 ± 0.9
Example B2	20	**2.87** ± **0.7**	**3.04** ± **0.6**	**3.10** ± **0.8**
Example B3	29	2.34 ± 0.6	2.78 ± 0.9	2.6 ± 1.2

Table 4.3: Experiment 2 Results. The highest-performing condition for each task is shown in bold. The highest-performing example-split condition is also shown in bold for each task.

	Mean Number of Newline Characters		
Example	Task A’	Task B’	Task C
Example A1	1.7	1.7	1.3
Example A2	2.3	2.2	2.0
Example A3	2.3	3.1	1.9
Example B1	*1.0*	*1.2*	*1.1*
Example B2	**3.0**	**3.6**	**2.3**
Example B3	2.4	2.5	2.0

Table 4.4: Average number of newline characters in reviews written by workers who saw each example

Exploratory Comparison of Examples

Recall that one of our goals was to see if there are differences in the efficacy of the various examples that workers saw, even though these examples all had comparably high quality. To examine this, we run a series of post-hoc comparisons. We note that the results reported here are exploratory and reported *p*-values do not indicate rigorous statistical significance. Nonetheless, we believe these analyses provide evidence that even among high quality examples, the pedagogical usefulness of the examples vary.

The bottom two sections of Table 4.3 show the results for the two good examples condition subdivided by each particular example. For example, a worker in the Example A1 row saw Example A1 as well as any of the three examples for Task B. We ran a post-hoc Kruskal-Wallis test for each task to determine if there were significant differences in performance based on the particular example that was seen from Task A; the test indicated no significant differences. However, a similar test for Task B seems to indicate that there are differences between the examples (p = 0.02 for Task A' and p = 0.06 for Task B'). To see which examples induce statistically significant improvements on review quality, we ran Mann-Whitney-U tests comparing workers who saw each particular example (coming from Task A or B) to the control, for each test task. For Task A', it appears that Example A2 and A3 had a positive effect on review quality ($p < 0.01$) and for all tasks, Example B2 had a positive effect on review quality ($p < 0.001$ for Task A' and $p < 0.02$ for Tasks B' and C). From Table 4.3, we can also see that Examples A1 and B1 in particular seem to be generally no better or possibly even worse than the control. Interestingly, Examples A2 and B2 are both not the highest rated examples. This suggests simply picking the highest quality example may have not led to as high improvements.

We now turn to see if the differences we see between the examples actually make sense. If we look at the content of the examples in Figures 4.3 and 4.4, we notice that Examples A1 and B1 both contain a single block of text (i.e., no spacing designating different paragraphs, although Example A1 does contain newline characters). Thus, while the content of these examples is not necessarily bad, the formatting may have a role in the effectiveness of the example. It is difficult to know what kind of a role formatting plays in workers' subsequent reviews, but it could potentially have an effect on cognitive load or where the workers' focus is drawn. A worker that merely skims a review might get more out of a review where a series of facts are quickly mentioned such as in the first paragraph of Examples A2 and B2 than a review that is front-loaded with dense text like Example A1.

In order to verify that the formatting actually has an effect on workers, we look at the average number of newline characters in reviews written by workers who saw each example, as shown in Table 4.4. Indeed, it appears that workers who see visually dense examples are more likely to write product comparison reviews with fewer newline characters. For example, workers who saw Example A1 or B1 used the least number of newline characters in their reviews, while workers who saw Example B2 included the most newline characters. While it is not clear if writing more newline characters is a proxy for writing a better review, it does seem clear that the examples have an impact on both the form and the quality of subsequent reviews. More work is needed to determine what factors contribute most to the pedagogical value of examples.

4.7 Discussion

As mentioned earlier, none of our initial hypotheses were met in the first experiment. Contrary to what we hypothesized, the one example condition did worse than control on the near transfer task (Task A') and seemingly not worse than control on the Task C (far transfer task). However, the results of our first experiment did seem to suggest that showing multiple examples from different categories of products appears to be more useful than seeing a single example. As mentioned earlier, we hypothesize that the one example condition performed poorly due to the quality of the examples. Indeed the median example quality from Task A that was shown to workers was 2.05 out of 5, which could explain why seeing even an above median example was not helpful. We cannot say whether two examples is really necessary or if a single good example (for instance, an example from Task B) is sufficient. However, our results do suggest that seeing a good example from the same category as the task to be completed by the worker does help. Thus, seeing multiple examples from different categories might be useful to provide more coverage of the type of tasks seen in the future. More work is needed to determine if seeing multiple good examples is better for far transfer than seeing a single good example.

In the second experiment, we did find statistically significant evidence for Hypothesis 1 (namely that training helps), but only for Task A', although the results suggest that seeing good examples helped on all three tasks. It could be that we did not have enough power to detect a significant effect on Tasks B' and C, both because there was a drop-out of workers after the first task and because the effect of examples might be lower on Tasks B' and C.

However, our results seem to suggest that if we had only shown a particular pair of examples to all workers (for instance, Examples A2 and B2), we may have seen a significant difference on Tasks B' and C. For example, the four workers who saw Examples A2 and B2 had an average performance of 3.67, which a post-hoc Mann-Whitney-U test suggests was significantly better than the control ($p = 0.01$) with a Glass' Δ effect of 1.2. These results seem to indicate that properly chosen examples can be pedagogically valuable for both near and far transfer tasks. Future experiments are needed to confirm this. This also suggests the potential benefit of using machine learning to try to automatically find the best example. One approach would be to use a multi-armed bandit to select the best example, perhaps after narrowing examples to ones that are of high quality, so that the algorithm would converge more quickly to finding a good example. This approach would be similar to the AXIS system [Williams et al., 2016]. However, unlike AXIS, our results suggest that we do not simply want to find examples that are rated as being of high quality, but rather, examples that actually lead to higher performance gains for workers.

Another approach is to fit a model that predicts how pedagogically valuable each example is. Prior work has examined fitting models to characterize the pedagogical value of crowdsourced examples and explanations [Aleahmad et al., 2010, Mustafaraj et al., 2018], but they have not actually used such models to select examples. Such a model could potentially generalize to predicting the pedagogical value of peer-generated examples that we have never tested on learners before. Additionally, such a model could predict what kinds of features make up a good peer-generated example, contributing to the learning sciences literature. A key challenge to this would be identifying the features of the examples that the model would use. Prior work has shown that simple features such as the length of the example could be a good indicator of pedagogical value [Doroudi et al., 2016]. Our second experiment suggests that possibly the presence of multiple paragraphs (newline characters) can also possibly serve as an indicator of how good a solution is. This agrees with prior work showing that clearly delineating sub-goals improves the efficacy of worked examples [Eiriksdottir and Catrambone, 2011]. Thus, it is possible that structural features of examples can be good proxies for whether the example is pedagogically valuable, but perhaps richer features that are based on the natural language of the examples could improve the accuracy of predictions. We plan to pursue such an approach to choosing examples in future work.

4.8 Conclusion

Taken together with the results of the last chapter, these results indicate that with adequate content curation, using peer work can be an effective way to generate new content for both problem solving tasks and open-ended writing tasks. However, the best way to curate content may vary from task to task. As such, future work should investigate whether domain agnostic machine learning methods can help filter peer content for different tasks. Curating content from large pools of learner-generated work can also have scientific value as it could help researchers better understand what constitutes effective pedagogical content for various tasks.

In Chapter 3, I examined the efficacy of validating peer work, and in this chapter, I examined the efficacy of presenting peer work as examples. I have shown that both could be effective, but have yet to perform a head-to-head comparison of these two modes of interacting with learnersourced content. Of course, each method has its advantages beyond learning gains. Validation can serve as a way to grade content, which may be necessary. (Indeed, even in this study we needed to have crowdworkers grade their peers' work to get quality metrics for each product review.) However, validation is more time consuming, so presenting peer work as examples may be a quicker way to get workers up to speed. Also, workers who are new to a task may not have enough prior knowledge to successfully validate peer work. However, future work should look into directly comparing these various modes of interacting with peer work, in addition to others, in order to develop a science of learnersourced curriculum design.

Part II

Instructional Sequencing

Chapter 5

Background: Mastery Learning and Reinforcement Learning

> What is the next step in the evolution of self-teaching devices? It would seem to be separation of the control and material presentation functions. The control function might logically be placed in a computer that has the memory and computational ability necessary to ascertain the student's knowledge and to learn and remember the student's learning characteristics...The teaching-learning process would now be a give-and-take procedure between student and machine similar to the student-teacher relationship.
>
> RONALD HOWARD, 1960

In 1960, a book was published by the name of *Dynamic Programming and Markov Decision Processes* and an article by the name of *Machine-Aided Learning*. The former established itself as one of the foundational early texts about Markov decision processes (MDPs), the model that underpins reinforcement learning (RL). The latter is a virtually unknown two-page vision paper suggesting that computers could help individualize the sequence of instruction for each student. Both were written by Ronald Howard, who is one of the pioneers of decision processes and is now considered the "father of decision analysis." These two lines of work are not unrelated; in 1962, Howard's doctoral student Richard Smallwood wrote his dissertation, *A Decision Structure for Teaching Machines*, on the topic of how to use decision processes to adapt instruction in a computerized teaching machine. This is perhaps the first example of using reinforcement learning (broadly conceived) for the purposes of instructional sequencing (i.e., determining how to adaptively sequence various instructional activities to help students learn). Instructional sequencing was thus one of the earliest applications of reinforcement learning. Since then a variety of attempts have been made to automatically determine how to sequence instructional activities for students.

In the next few chapters, I will investigate two broad types of automating instructional sequencing. Cognitive mastery learning[1] [Bloom, 1968, Corbett, 2000] is a standard approach to sequencing instructional practice that is used in many ITSs. Cognitive mastery learning typically assumes a model of student learning, such as Bayesian knowledge tracing (BKT), and provides students with practice on each knowledge component until the student is believed to have reached mastery for that knowledge component. An assumption made in cognitive mastery learning is the knowledge decomposition hypothesis—that knowledge can be decomposed into parts that can be learned independently once all prerequisite knowledge is learned [Corbett, 2000]. Determining how much practice to give on each knowledge component is a simple form of instructional sequencing. To tackle broader forms of instructional sequencing, one can use reinforcement learning to find an instructional policy (a method of sequencing problems that is often adaptive to some student state). While reinforcement learning approaches to instructional sequencing date back to the 1960s when Markov decision processes first emerged, they have recently regained popularity with modern advances in reinforcement learning.

The key point in this part of my dissertation is that various approaches to instructional sequencing, such as cognitive mastery learning and reinforcement learning, lie on a bias-variance tradeoff. The bias-variance tradeoff is an important concept in machine learning and statistics [Geman et al., 1992] that refers to the fact that when trying to make a statistical prediction (e.g., estimating a parameter of a distribution or fitting a function), there is a tradeoff between the accuracy of the prediction (bias) and its precision (variance). This is perhaps best understood in the context of machine learning algorithms that must fit a function using a *model class*. A model class is a set of models that usually have the same underlying form, but must be instantiated with particular parameters. For example, the class of linear estimators using a particular set of features in linear regression is one model class. The bias of a model class represents how different the best fitting model in the model class is from the target function. For example, if we wanted to predict y where $y = 3.5x^2 - x$ using a linear estimator, we could never fit the curve perfectly, and thus, the model would have bias. On the other hand, if we were to use a model class of quadratic estimators (i.e., all functions of the form $y = ax^2 + bx + c$, then we could fit the target perfectly, and so the model would have no bias. Variance represents the amount of variability in models of the model class, or in other words is a measure for the complexity or size of the model class. The model class of quadratic estimators has higher variance than linear estimators because it has more degrees of freedom.

To make accurate predictions, one must try to find a model class that effectively *balances* the bias-variance tradeoff. Figure 5.1 shows various approaches to instructional sequencing that I discuss in my thesis and how they lie on the bias-variance tradeoff. I will refer to this diagram throughout the next few chapters, both to discuss the limitations of some of these approaches, as well as to introduce ways of navigating this tradeoff. In this rest of this chapter, I provide background on mastery learning and reinforcement learning applied to instructional sequencing.

[1] I will often use the terms cognitive mastery learning and mastery learning interchangeably, although technically cognitive mastery learning specifically refers to cases where mastery learning involves some cognitive model of student learning to contrast it with other approaches to mastery learning that were used in pen-and-paper settings [Bloom, 1968] as well as mastery learning approaches that use heuristics such as three-consecutive-correct-in-a-row [Kelly et al., 2016].

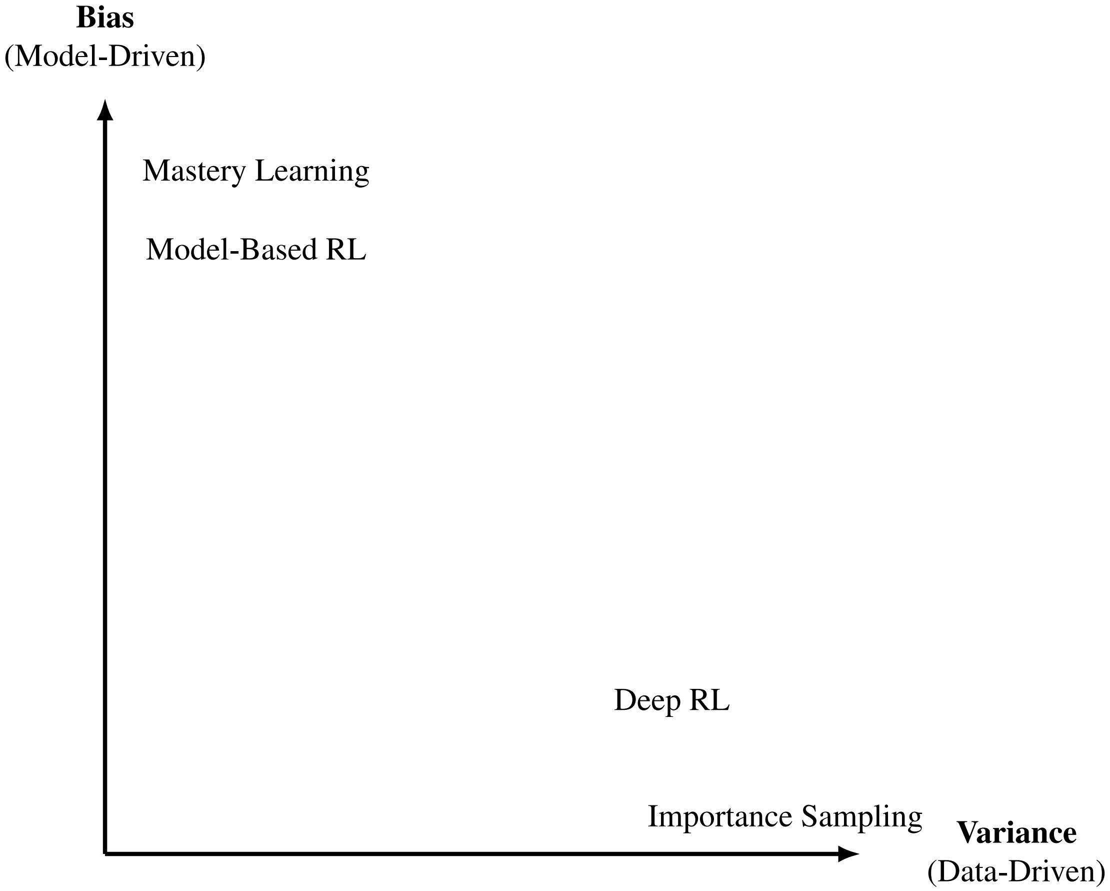


Figure 5.1: The bias-variance tradeoff in approaches to instructional sequencing. At the upper left are approaches that are more model-driven (i.e., make stronger assumptions about how people learn), and at the bottom right are more data-driven approaches (i.e., make less assumptions about how people learn and make more data-driven inferences). The more model-driven techniques are biased and are prone to under-fitting, while the more data-driven approaches have high variance and are prone to over-fitting. The ideal would be to find a method that is closer to the origin of the axes. (The positioning of these methods is meant for illustrative purposes only and is by no means meant to precisely capture the bias and variance of the various methods.)

5.1 Mastery Learning

Knowledge tracing algorithms are used in learning technologies such as intelligent tutoring systems [Corbett and Anderson, 1995, Ritter et al., 2007], massive open online courses [Rosen et al., 2018], and Khan Academy [Hu, 2011], in order to adaptively assess learners' knowledge states and use that assessment to implement mastery learning (i.e., decide when students have mastered skills and are ready to move on to other skills). We assume that for each skill, students are given a number of practice opportunities for that skill and on each practice opportunity the student will give a response that is either correct or incorrect. The goal of a knowledge tracing algorithm when used for mastery learning is to determine when to stop giving students practice opportunities for the given skill. Knowledge tracing is often performed by a statistical model of student learning that could be fit to data. I will describe two such model classes and how they can be used to implement mastery learning. In addition to model-based knowledge tracing, simple heuristics can be used to implement mastery learning such as the N-Consecutive Correct Responses (N-CCR) heuristic, which simply gives practice opportunities until the student answers N questions correctly in a row. For simplicity, throughout this chapter and Chapters 6 and 7, I assume students are only learning a single skill, but all the ideas can be extended to learning multiple skills.

5.1.1 Bayesian Knowledge Tracing (BKT)

The BKT model is the most commonly used model for knowledge tracing [Corbett and Anderson, 1995, Ritter et al., 2007, Rosen et al., 2018]. The Bayesian Knowledge Tracing model is a two-state hidden Markov model (HMM) that assumes for each skill, that the student either knows the skill or they do not. At each practice opportunity $i \geq 1$ (i.e., when a student has to an answer a question corresponding to the skill), the student has a latent knowledge state $K_i \in \{0, 1\}$. If the knowledge state is 0, the student has not learned the skill, and if it is 1, then the student has learned it. The student's answer can either be correct or incorrect: $C_i \in \{0, 1\}$ (where 0 corresponds to incorrect and 1 corresponds to correct). Students initially know the skill with probability $P(L_0) = P(K_0 = 1)$. With every practice opportunity, students have some probability of learning the skill if they do not already know it ($P(T)$). If the student does not know the skill, the student will guess the correct answer with probability $P(G)$ and if the student does know the skill, the student will answer correctly unless they slip with probability $P(S)$. These four parameters fully describe the standard BKT model for each skill. The Bayesian knowledge tracing algorithm proceeds by maintaining a probabilistic belief that the student has learned the skill given the sequence of observed responses and the parameters of the BKT model. Once this probability exceeds some mastery threshold (typically taken to be 0.95), the algorithm assumes the student has mastered the skill.

One can learn the parameters of the BKT model by fitting it to a dataset consisting of sequences of student responses. In Chapter 6, we fit the BKT models using brute-force grid search over the entire parameter space in 0.01 increments with the BKT Brute Force model fitting code [Baker et al., 2010a].

5.1.2 Additive Factor Model (AFM)

AFM is another popular model of student learning, but it is not typically used to perform knowledge tracing; rather, AFM is often used to examine learning curves after students go through mastery learning [Cen, 2009]. Unlike BKT, AFM assumes that learning takes place incrementally; with each practice opportunity, the probability that the student will answer correctly on future practice opportunities increases. This also means that unlike BKT, AFM does not try to predict a latent knowledge state (whether the student has learned the skill or not), but rather directly models student performance (the probability of answering correctly at any given time). In particular, a simplified version of AFM for when there is only one skill is governed by the following logistic function:

$$P(C_i = 1) = \frac{1}{1 + \exp(-(\theta - \beta + \gamma i))}$$

where $P(C_i = 1)$ is the probability that the student will answer the ith practice opportunity correctly, θ is the student ability (which could encapsulate the student's prior knowledge), β is the difficulty of the skill, and γ is the learning rate. Since we are only interested in a single skill, we set $\beta = 0$ and let θ combine the student ability and the item difficulty. Notice that another difference between AFM and BKT is that AFM allows for individual differences via θ. Thus, if we want to use AFM for knowledge tracing, assuming we have not interacted with a particular student before, we would need to estimate θ online (as is done in computerized adaptive testing). When using AFM to implement mastery learning, we need to choose a certain level of desired accuracy for the mastery threshold.

5.2 Reinforcement Learning: Towards a "Theory of Instruction"

In 1972, the psychologist Richard Atkinson wrote a paper titled *Ingredients for a Theory of Instruction* [Atkinson, 1972b], in which he claims a theory of instruction requires the following four "ingredients":

1. A model of the learning process.
2. Specification of admissible instructional actions.
3. Specification of instructional objectives
4. A measurement scale that permits costs to be assigned to each of the instructional actions and payoffs to the achievement of instructional objectives.

Atkinson further describes how these ingredients for a theory of instruction map onto the definition of a Markov decision process (MDP). Formally, a finite-horizon MDP [Howard, 1960a] is defined as a five tuple (S, A, T, R, H), where

- S is a set of states
- A is a set of actions
- T is a transition function where $T(s'|s, a)$ denotes the probability of transitioning from state s to state s' after taking action a

- R is a reward function where $R(s, a)$ specifies the reward (or the probability distribution over rewards) when action a is taken in state s, and
- H is the horizon, or the number of time steps where actions are taken.

In *reinforcement learning* (RL), the goal is for an *agent* to learn a policy π that specifies the action to take in each state (or a probability distribution over actions) that incurs a large reward [Sutton and Barto, 1998]. There exist various methods for *planning* in a MDP, such as value iteration [Bellman, 1957] or policy iteration [Howard, 1960a], which yield the optimal policy for the given MDP. However, RL refers to the task of learning a policy when the parameters of the MDP (the transition function and possibly the reward function) are not known ahead of time.

As Atkinson explained, in the context of instruction, the transition function maps on to a model of the learning process, where the MDP states are the states that the student can be in (such as cognitive states). The set of actions are instructional activities that can change the student's cognitive state. These activities could be problems, problem steps, flashcards, videos, worked examples, or game levels in the context of an educational game. Finally, the reward function can be factorized into a cost function for each instructional action (e.g., based on how long each action takes) and a reward based on the cognitive state of the student (e.g., a reward for each skill a student has learned). As we show below, the natural formulation of the instructional process as a decision process and a problem that can be tackled by reinforcement learning drew many researchers, including psychologists like Atkinson, to this problem. In theory, RL could formalize that which was previously an art: instruction. How well it can do so in practice is the subject of investigation in Chapter 9.

5.2.1 Examples of RL for Instructional Sequencing

In order to better situate how RL is used for instructional sequencing, it is worth giving some concrete examples of how the techniques of decision processes and RL could be applied to instructional sequencing. We will begin with one of the simplest possible MDPs that could be used in the context of instructional sequencing, and then consider a series of successive refinements to the model to be able to model more authentic phenomena, ending with the model considered by Atkinson [1972b]. While there are many more ways of applying RL to instructional sequencing, this section will give us a sense of one concrete way in which it has been done, as well as introduce several of the design decisions that need to be made in modeling how people learn and using such models to induce instructional policies. In the review of empirical studies below, we will discuss a much broader variety of ways in which various researchers have used RL to implement instructional sequencing.

The first model we will consider is a simple MDP that assumes for any given fact, concept, or skill to be learned (which we will refer to as a knowledge component or KC), the student can be in one of two states: the "correct" state or the "incorrect" state. Whenever the student answers a question correctly, the student will transition to the correct state for that the associated KC, and whenever the student answers a question incorrectly, the student will transition to the incorrect state for that KC. The complete state can be described with a binary vector of all the individual KC states. The set of actions is the set of items that we can have students practice, where each item is associated with a given KC. For each item, there is a 2-by-2 transition matrix that specifies the probability of its associated KC transitioning from one state to another. (For simplicity, we assume that all items for the same KC have the same probability of transitioning to the learned state.) Suppose our goal is to have the student reach the correct state for as many KCs as possible. Then we can specify a reward function that gives a reward of one whenever the student transitions from the incorrect state to the correct state, a reward of negative one whenever the student transitions from the correct state to the incorrect state, and a reward of zero otherwise. In this case, the optimal instructional policy is trivial: always give an item for the KC that has the highest probability of transitioning to the correct state among all KCs in the incorrect state.

Of course to use this policy in practice, we need to learn the parameters of the MDP. We can learn the maximum likelihood transition parameters using data collected from prior students. Given the assumptions we made, the only parameters in this model are the transition probabilities for each KC. In this case, the maximum likelihood transition probability can be inferred simply by computing how many times students transitioned from the unlearned state to the learned state divided by the number of time steps where the students were in the unlearned state.

However, notice that the MDP presented above is likely not very useful if students have some chance of guessing questions correctly, because then a student might answer correctly without really understanding a KC. In reality, we may assume that students' answers are only noisy signals of their underlying knowledge states. To model this, we would need to use a partially observable Markov decision process (POMDP) [Sondik, 1971]. In a POMDP, the underlying state is inaccessible to the agent, but there is some observation function (O) which maps states to potential observations. In our example, the observation at each time step is whether the student answers a question right or wrong and the probability of answering a question right or wrong depends on whether the student is in the learned state or unlearned state for the current KC that is being taught. If we ignore the reward function, this POMDP is equivalent to the BKT model. Typically BKT is not considered in the RL framework, because a reward function is not *explicitly* specified, although using BKT for mastery learning does *implicitly* follow a reward function. One possible reward function for mastery learning would be that each time our estimated probability that the student has learned a particular KC exceeds 0.95, then we receive a reward of one, and otherwise we receive a reward of zero. Such a model would then keep giving items for a given KC, until we are 95% confident that the student has learned that KC before moving on. Notice that the optimal policy under this reward function (i.e., cognitive mastery learning) is very different from the optimal policy under a different reward policy (e.g., get a reward of one for each KC that is *actually* learned).

The parameters of a POMDP like the BKT model are slightly more difficult to infer, because we do not actually know when students are in each state, unlike in the MDP case. However, there are number of algorithms that could be used to estimate POMDP parameters including expectation maximization [Welch, 2003], spectral learning approaches [Falakmasir et al., 2013, Hsu et al., 2012], or simply performing a brute-force grid search over the entire space of parameters [Baker et al., 2010b].

We consider one final modification to the model above, namely that which was used by Atkinson

[1972b] for teaching German vocabulary words. Note that the BKT model does not account for forgetting. Atkinson [1972b] proposed a POMDP with three states for each word to be learned (or KC, in the general case): an unlearned state, a temporarily learned state, and a permanently learned state. The model allows for some probability of transitioning from either the unlearned state or the temporarily learned state to the permanently learned state, but one can also transition from the temporarily learned state back to the unlearned state (i.e., forgetting). Moreover, this model assumes that a student will always answer an item correctly unless the student is in the unlearned state, in which case the student will always answer items incorrectly. The reward function in this case gives a reward of one for each word that is permanently learned at the end (as measured via a delayed posttest, where it is assumed that any temporarily learned word will be forgotten). The optimal policy in this case can be difficult to compute because one needs to reason about words that are forgotten over time. Therefore, Atkinson [1972b] used a myopic policy that chooses the best next action as though only one more action will be taken. In this case, the best action is to choose the word that has the highest probability of transitioning to the permanently learned state.

5.2.2 Design Considerations in Reinforcement Learning

Before continuing, it is worthwhile to describe several different settings that are considered in reinforcement learning, and the design considerations that researchers need to make in considering how to apply RL. RL methods are often divided into *model-based* and *model-free* approaches. Model-based RL methods learn the model (transition function and reward function) first and then use MDP planning methods to induce a policy. Model-free methods use data to learn a good policy directly without learning a model first. Most of the studies that have used RL for instructional sequencing have used model-based RL. All of the examples described above are model-based—a model is fit to data first and then a policy (either the optimal policy or a myopic policy) is derived using MDP/POMDP planning.

There are two different ways in which RL can be used. In *online* RL, the policy is learned and improved as the agent interacts with the environment. In *offline* RL, a policy is learned on data collected in the past, and is then used in an actual environment. For instance, in the examples we presented above, the models were fit to previously collected data in an offline fashion, which was then used to do model-based RL.

In online RL, the agent must decide whether to use the current best policy in order to accrue a high reward or to make decisions which it is uncertain about with the hopes of finding a better policy in the future. This is know as the exploration vs. exploitation trade-off. Exploration refers to trying new actions to gather data from less known areas of the state and action space, while exploitation refers to using the best policy the agent has identified so far. This trade-off has rarely been considered in applying RL to instructional sequencing.

As discussed in our examples, since the cognitive state of a student usually cannot be observed, it is common to use a partially observable Markov decision process rather than a (fully observable) MDP. Planning, let alone reinforcement learning, in POMDPs is, in general, intractable, which is why researchers often use approximate methods for planning, such as myopic planning. However, some models of learning (such as the BKT model discussed above) are very restricted POMDPs, which makes it possible to find an optimal policy.

In model-based RL, our model is generally incorrect, not only because there is not enough data to fit the parameters correctly, but also because the form of the model could be incorrect. As we will see, researchers have proposed various models for student learning, which make rather different assumptions. When the assumptions of the model are not met, we could learn a policy that is not as good as it seems. To mitigate this issue, researchers have considered various methods of *off-policy policy evaluation*, or evaluating a policy offline using data from one or more other policies. Off-policy policy evaluation is important in the context of instructional sequencing, because it would be useful to know how much an instructional policy will help students before testing it on actual students. Ultimately, a policy must be tested on actual students in order to know how well it will do, but blindly testing policies in the real world could be costly and potentially a waste of student time.

From the intelligent tutoring systems literature, we can distinguish between two broad forms of instructional sequencing in terms of the granularity of the instructional activities: *task-loop* (or outer loop) adaptivity and *step-loop* (or inner loop) adaptivity [Aleven et al., 2016a, Vanlehn, 2006, VanLehn, 2016]. In task-loop adaptivity, the RL agent must select distinct tasks or instructional activities. In step-loop adaptivity, the RL agent must choose the exact sequence of steps for a fixed instructional task. For example, an RL agent operating in the step loop might have to decide for all the steps in a problem whether to show the student the solution to the next step or whether to ask the student to solve the next step [Chi et al., 2009]. While step-loop adaptivity is a major area of research in adaptive instructional sequencing in general [Aleven et al., 2016a], relatively little work has been pursued in this area using RL-based approaches.

Finally, I note that in this thesis, I specifically focuses on applications of reinforcement learning to the sequencing of instructional activities. Reinforcement learning and decision processes have been used in other ways in educational technology that we do not consider here. For example, Barnes and Stamper [2008] have used MDPs to model students' problem solving processes and automatically generate hints for students. Similarly, Rafferty et al. [2015b, 2016b] modeled student problem solving as a MDP and used problem solving trajectories to infer the MDP so they could ultimately give feedback to the students about misconceptions they might have. In these papers, the actions of the MDP are problem solving steps *taken by students* in the course of solving a problem, whereas in our paper, we focus on studies where the actions are instructional activities *taken by an RL agent* to optimize a student's learning over the course of many activities.

Chapter 6

The Bias of Bayesian Knowledge Tracing[*]

All models are wrong but some are useful

GEORGE BOX, 1979

The key point I explore in the following two chapters is the idea that our statistical models of student learning are not accurate representations of how students learn. This point is not contentious; psychologists and neuroscientists do not actually believe that students acquire complex skills according to the Bayesian knowledge tracing model (especially when students are assumed to never forget what they have learned) [MacLellan et al., 2016]. The question that remains is, are such models accurate enough to be effective in driving instructional sequencing? To explore this issue we explicitly consider model misspecification: what happens if student learning is actually governed by a different model of learning than the particular statistical student model that we choose to use to model it? I specifically look at how model misspecification can lead to misguided conclusions about the BKT model. Again, most researchers and practitioners realize that BKT is likely not an accurate model of student learning—and therefore model misspecification is not a foreign concept—but I content that we often use student models (like BKT) and interpret them as though they are "correct" models. According to the famous quantitative sociologist, Otis Dudley Duncan [1975]: "there are many more wrong models than right ones, so that specification error is very common, though often not recognized and not easily recognizable."

The overall argument I use in both this chapter and the next is as follows. First I describe a problem noticed by researchers when using the BKT model. I then show how this problem can potentially be explained in terms of model misspecification. Finally, I discuss the consequences of using such a misspecified BKT model for mastery learning. The general methodology I use was aptly explained by Duncan [1975]:

> Analysis of specification error relates to a rhetorical strategy in which we suggest

[*] The work described in this chapter was largely adapted from Doroudi and Brunskill [2017].

> a model as the "true" one for sake of argument, determine how our working model differs from it and what the consequences of the difference(s) are, and thereby get some sense of how important the mistakes we will inevitably make may be.

Therefore, when I propose "a model as the 'true' one," it is important to note that I am doing so "for sake of argument", and by no means am I claiming such a model is the true model of student learning or even a more correct model than BKT.

The particular argument I make in this chapter is as follows. I first present the problem that BKT models that are fit to data often have unrealistic guess and slip parameters. Prior work has tried to explain this as a result of identifiability. However, I have shown that BKT does not suffer from an identifiability problem as was previously thought [Doroudi and Brunskill, 2017]. Rather, I show that these parameters might be due to model misspecification. Namely, if learning is actually more of a gradual process than the all-or-nothing assumption that BKT makes, then the fitted BKT model might have parameters that are semantically implausible. I then show how using such a BKT model for mastery learning can lead to incorrectly assuming many students have mastered a skill before they actually have. In the following chapter, I examine how using a misspecified BKT model might lead to giving less practice than needed to under-performing students, hence suggesting that mastery learning could be inequitable when using the wrong model.

By highlighting these concerns with using BKT, my point is not to suggest that there is some other model that is unbiased; rather my point is that when using simple models of student learning for instructional sequencing, we need to understand how robust those models are to model misspecification. Chapter 8 will then look at how we can use multiple models of learning to robustly identify good instructional policies.

6.1 Semantic Model Degeneracy

Problem: Several researchers have found that when fitting a BKT model to data, the model might have parameters that are not semantically plausible. For example, Baker et al. [2008] found that for 75% of skills in their middle school mathematics tutoring system, either $P(G) > 0.5$ or $P(S) > 0.5$, which is hard to reconcile with our intuitive notions of guessing and slipping, which suggest that they should be infrequent events.

Explaining the Problem: Beck and Chang [2007] proposed that this is due to an *identifiability problem*, where multiple sets of parameters can explain the data equally well. However, I have shown using prior results on the identifiability of hidden Markov models that BKT (except in some technically degenerate cases) is an identifiable model [see Doroudi and Brunskill, 2017].

Contribution: Since unidentifiability cannot explain semantically implausible BKT parameters, I instead claim that this problem of *semantic model degeneracy* might be explained by model misspecification. That is, when BKT does not accurately capture student learning, the resulting model fit might be semantically degenerate. To better understand this, I first classify the various types of semantic model degeneracy. I then show how model misspecification can result in semantically degenerate BKT models and how that could have adverse consequences when the models are used for mastery learning.

6.1.1 Types of Semantic Model Degeneracy

Baker et al. [2008] distinguish between two forms of semantic model degeneracy: *theoretical degeneracy* and *empirical degeneracy*. They define a model to be theoretically degenerate when either the guess or the slip parameter is greater than 0.5. They define a model to be empirically degenerate if one of two things occur: (1) for some large enough n the model's estimate of the student having mastered the skill decreases after the student gets the first n skills correct or (2) for some large enough m, the student does not achieve mastery (our estimate of the student having learned the skill does not go beyond 0.95) even after m consecutive correct responses [Baker et al., 2008]. The authors chose the values $n = 3$ and $m = 10$. The first form of empirical degeneracy is only possible if $1 - P(S) < P(G)$ (i.e., the student is more likely to answer a question correctly if they have not learned the skill than if they have learned the skill), as was shown by van De Sande [2013]. This is true, even for $n = 1$. Thus, this first notion of empirical degeneracy is equivalent to $P(G) + P(S) > 1$, which implies either $P(S) > 0.5$ or $P(G) > 0.5$, meaning that it always implies theoretical degeneracy! Huang et al. have noted that while $P(G) + P(S) > 1$ implies semantically degenerate parameters as it contradicts mastery, the condition that $P(G) < 0.5$ and $P(S) < 0.5$ may not always be necessary for the parameters to be semantically meaningful, since, for example, there may be some domains where the student can guess the correct answer easily [Huang et al., 2015]. We agree that suggesting $P(G) < 0.5$ is degenerate does seem somewhat arbitrary depending on the domain; however, we do think $P(S) > 0.5$ should be characterized as a form semantic degeneracy, because, as [Baker et al., 2008] claimed, it does not make sense for a student who has learned a skill to answer questions of that skill incorrectly most of the time—that goes against our intuitions of what mastery means. Given the limitations of empirical and theoretical degeneracy, we find it more useful to categorize the forms of semantic model degeneracy by what they suggest about student learning:

- *Forgetting*: This occurs when $P(G) + P(S) > 1$, which suggests that not only are students not learning, but that students are more likely to lose knowledge of a skill as they receive more practice opportunities for that skill. Another way to view this degeneracy is that the state we would conceptually call the learned knowledge state is now the state where performance is worse.

- *Low Performance Mastery*: This occurs when $P(S) > 0.5$. We can also set a lower threshold for low performance mastery (e.g., $P(S) > 0.4$).

- *High Performance Guessing*: This occurs when $P(G) > t$, where t is some threshold. As mentioned earlier, this seems like a weak form of degeneracy, as students can often guess an answer easily even if they have not learned a skill, but we can set t to a large enough value, to make this a form of model degeneracy.

	State i									
	0	1	2	3	4	5	6	7	8	9
$P(K_0 = k)$	0.1	0.1	0.1	0.2	0.2	0.3	0	0	0	0
$P(C_i = 1 \mid K_i = k)$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$P(K_i = k + 1 \mid K_i = k)$	0.4	0.3	0.2	0.1	0.05	0.05	0.05	0.05	0.05	-

Table 6.1: Alternative model of student learning where there are ten levels of mastery

	10-State HMM		AFM	
# of Practice Opportunities	20	200	20	200
$P(L_0)$	0.30	0.001	0.09	0.001
$P(T)$	0.05	0.02	0.05	0.05
$P(G)$	0.27	0.49	0.14	0.28
$P(S)$	0.44	0.13	0.46	0.03

Table 6.2: BKT models fit to data generated from the model described in Figure 6.1 and an additive factors model described in the text. The first column for each model is fit to 500 sequences of 20 practice opportunities, while the second column is fit to 100 sequences of 200 practice opportunities. The models were fit using brute-force grid search over the entire parameter space in 0.01 increments for the parameters using the BKT Brute Force model fitting code [Baker et al., 2010a].

- *High Performance ⇏ Learning*: This is the second form of empirical degeneracy given by [Baker et al., 2008]: for some choice of m, the probability that the student has learned a skill is less than some threshold p (typically taken to be 0.95) after m consecutive correct responses

6.2 Model Misspecification and Semantic Model Degeneracy

We will now consider a possible explanation for why BKT models can result in semantic model degeneracy (which we believe to be part of the reason that researchers look towards identifiability and local optima to explain the strange parameters that result from fitting BKT models). First of all, note that forgetting degeneracy will occur whenever students actually do forget or when they learn misconceptions; it is not unreasonable to believe that students will sometimes learn and reinforce a misconception, causing their knowledge of some skill to decrease over time. Thus, while this form of degeneracy technically violates our notion of mastery, it is to be expected if we switch the semantic interpretation of the two states and suppose that students forget instead of learn. We now consider sources of the other forms of semantic model degeneracy.

We claim that such forms of semantic model degeneracy can result from not accurately being able to capture the complexity of student learning with a two state HMM. When this is the case, fitting the data with a two state HMM will result in trying to find the best fit of the data for a two state HMM. However, this may result in a model that does not align with our intuitions about what it means for a student to have learned a skill.

To support our claim, suppose student learning is actually governed by a 10-state HMM with ten consecutive states representing different *levels* of mastery. From each state, the student has some probability of transitioning to the next state (slightly increasing in mastery), and from each state, the student has a probability of answering questions correctly. This probability strictly increases as the student's level of mastery increases. Specifically consider the model presented in Table 6.1[1]. Now suppose we try to use a standard BKT model to fit data generated from this alternative model of student learning. The first two columns of Table 6.2 show the parameters of BKT models fit to 500 sequences of 20 practice opportunities or 100 sequences of 200 practice opportunities, both generated from the the model in Table 6.1.

Notice that the model fits (nearly) degenerate parameters in both cases. When we only have 20 observations per student, the model estimates a very high slip parameter. This is because it has to somehow aggregate the different latent states which correspond to different levels of mastery, and since not many students would have reached the highest levels of mastery in 20 steps, it is going to predict that students who have "learned" the skill are often getting it wrong. However, what's more interesting is that for the same model, if we simply increase the number of observations per student from 20 to 200, we find that the slip parameter is reasonably small, but now the guess probability is 0.49. This is because, by this point most students have actually reached the highest level of mastery, so to compensate for the varying levels of mastery that occurred earlier in student trajectories, the model will have to estimate a high guess parameter. So we find that not only can alternative models of student learning lead to fitting (near) degenerate parameters, but varying the number of observations can lead to different forms of degeneracy. This is a counterintuitive phenomenon that we believe is not the result of having insufficient data (students) to fit the models well, but rather the result of the mismatch between the true form of student learning and the model we are using the fit student learning.

We find similar results if we fit a BKT model to data generated from another alternative model of student learning that is commonly used in the educational data mining community, the additive factor model (AFM) [Cen, 2009]. In particular, we used the model

$$P(C_i = 1) = \frac{1}{1 + \exp(-\theta + 2 - 0.1i)}$$

where $\theta \sim \mathcal{N}(0, 1)$ is the student's ability[2]. The second two columns of Table 6.2 show the parameters of BKT models fit to data generated from this model. We again find that when using

[1] Recall that we are proposing the 10-state HMM for sake of argument, not to actually suggest that this is a more reasonable model for how students learn. In particular, our motivation for using such a model is that it suggests students learn more gradually than the BKT model, and as we show, this could lead to semantic model degeneracy. We show below that we obtain similar results if learning were governed by the additive factor model, another more incremental model of learning. Thus, we believe these results generally hold when learning happens more incrementally than suggested by BKT.

[2] This model suggests that students who are two standard deviations above the mean initially will answer correctly half the time, and after 20 practice opportunities the average student will answer correctly half the time.

only data with 20 practice opportunities, we fit a high slip parameter, but when we using data with 200 practice opportunities, we fit a higher guess parameter and a very small slip parameter.

These results collectively suggest that if the assumption that learning is all-or-nothing (or that students learn a skill instantaneously with some probability) is incorrect, fitted BKT models could be semantically degenerate. Of course, this is only one form of statistical model misspecification. While it seems reasonable that at least in some cases learning is more incremental than BKT suggests, we cannot conclude that this is always the reason that BKT models have semantically degenerate guess and slip parameters. In particular, another form of model misspecification, namely KC model misspecification may also lead to semantically degenerate parameters. KC model misspecification occurs when it is incorrectly assumed that multiple items share or do not share the same knowledge component. KC model misspecification and various techniques for mitigating it have been studied much more, to our knowledge, than statistical model misspecification, although typically in the context of AFM, not BKT [Koedinger et al., 2012a, Stamper and Koedinger, 2011]. In reality, it is likely that both forms of model misspecification occur in real-world settings, and as such we must be aware of the consequences of each (and how they might interact). In this thesis, I only consider BKT models for a single skill, and as such it only makes sense to consider statistical model misspecification. Future work should consider the interplay between the two.

6.3 Model Misspecification and Mastery Learning

These observations have important implications for how learned models might be used in automated sequencing of content, such as cognitive mastery learning. Using such a BKT model to predict student mastery can lead to problematic inferences. For example, for the first model in Table 6.2, the BKT model assumes that when a student has reached mastery, they have a 56% chance of answering a question correctly, whereas a student who has actually mastered the skill will have a 90% chance of answering correctly (see Table 6.1). Thus, an intelligent tutoring system that uses such a BKT model to determine when a student has had sufficient practice on a problem, will likely give far fewer problems to the student than they actually need in order to reach mastery. To illustrate this, Table 6.3 shows the average number of practice opportunities the first model in Table 6.2 will give, when students actually learn according to Table 6.1. In contrast, the average number of practice opportunities needed to reach mastery according to the true model is around 100. Thus, cognitive mastery learning could lead to a significant amount of under-practice, even with a very high mastery threshold (e.g., 0.9999). This case study provides an example of how reasoning about model mismatch can be informative in terms of understanding the instructional consequences of our models.

Mastery Threshold	Avg. # Opportunities to Mastery	% Students with Under-Practice
0.95	28.4	99.4%
0.99	38.3	99%
0.9999	53.4	95%

Table 6.3: The average number of practice opportunities needed for the model given in the first column of Table 6.2 to reach mastery for various mastery thresholds, given that the true model is the model from Table 6.1. Averages were taken over 500 simulated students. The third column shows the percentage of simulated students that received less practice than needed. In contrast, the average number of practice opportunities that it took simulated students to reach mastery was around 100.

6.4 Discussion

If the analysis above suggests that BKT is the wrong model and that this can have problematic consequences for mastery learning, what should we do to mitigate these concerns? One option is to continue our quest for finding more accurate student models. However, we must acknowledge that models that have higher predictive accuracy are not necessarily more correct. For example, two models may have similar predictive accuracy while making drastically different predictions about how students learn (such as AFM and BKT). In this case, the two models may capture various competing aspects of learning, but miss other aspects (since learning is a complex, multi-faceted phenomena). Moreover, even if we knew the true form of the model to capture student learning, if that model has many parameters it could require inordinate amounts of data to fit; thus with finite data, a simple model such as BKT may actually result in better model fits than the “true model.” Therefore, while finding models with higher predictive accuracy can help in improving the quality of our student models, we should proceed with caution. Moreover, the work above suggests additional metrics we can use in determining how good a model is beyond standard metrics for predictive accuracy. For example, we can see if the parameters of our models are semantically plausible, and moreover if the parameters are stable when we change the number of practice opportunities used to fit the models. While model fit should improve when we have more training data, the parameters should not vary drastically (e.g., changing from high $P(S)$ to high $P(G)$).

If our ultimate goal in using student models is a pragmatic goal of improving student learning rather than a scientific goal of fully understanding and accurately modeling how people learn, then perhaps we do not need to get caught up in a quest for the perfect student model. Instead, we should search for models that are effective towards the ends that are being used for, such as accurately predicting when students have mastered skills. We will revisit how to do this to some extent in the next chapter and moreso in Chapter 8. For now, suffice it to say that perhaps we should not be searching for correct models, but rather for useful ones; after all, “all models are wrong but some are useful” [Box, 1979].

Chapter 7

The Equitability of Bayesian Knowledge Tracing[*]

> Educational equity means that each child receives what he or she needs to develop to his or her full academic and social potential.
>
> National Equity Project

A major challenge in any learning environment is ensuring that students with different needs receive personalized instruction to suit those needs. If instruction is not individualized, then some students may lag behind, while others proceed at a pace that is slower than ideal. Adaptive educational technology is designed to alleviate some of these concerns by giving students instruction and practice at the pace they need it. While this means that some students may need to work longer than others, ideally all students will eventually be able to master the content at hand. In practice, however, adaptive technologies sometimes fall short of this goal, with lower-performing students receiving less practice and instruction than they need. As some researchers have pointed out, educational technologies that aim to benefit all learners might disproportionately benefit more advantaged groups of learners [Hansen and Reich, 2015, Reich and Ito, 2017].

In this chapter, I examine the equitability of knowledge tracing algorithms that implement mastery learning, following the same methodology as in the previous chapter. I first show that the adaptivity provided by knowledge tracing makes it substantially more equitable than providing all students with the same amount of instruction. However, I show that knowledge tracing algorithms can still be inequitable (favoring fast learners over slow learners or high prior ability learners over low prior ability learners) when they rely on inaccurate models of student learning. In particular, I show that this issue could arise in two situations: (1) when using a BKT model that is fit to aggregate populations of students, and (2) when students learn more incrementally than suggested by the BKT model (i.e., when students learn according to AFM, as studied in the previous chapter). Moreover, I show that using AFM for mastery learning may lead to more equitable outcomes, even under model misspecification. Using AFM for mastery learning has

[*] The work described in this chapter was largely adapted from Doroudi and Brunskill [2019].

some limitations, so our results demonstrate the need for more work in developing robustly equitable knowledge tracing algorithms. The following chapter will present a general methodology for evaluating instructional policies (beyond just mastery learning policies) to ensure they are robust to model misspecification, which can also be applied to ensuring instructional policies are equitable. The broader message of this chapter is that when designing learning analytics algorithms, we need to explicitly consider whether the algorithms act fairly with respect to different populations of students, and if not, how we can make our algorithms more equitable.

7.1 Equity of Mastery Learning

Problem: Even though mastery learning aims to help each student master all skills, several researchers have found that using BKT for mastery learning sometimes leads to worse outcomes for some learners than others. For example, Corbett and Anderson [1995] found the following:

> The model underestimates the true learning and performance parameters for above-average students. As a result, these students who make few errors receive more remedial exercises than necessary and perform better on the test than expected. In contrast, the model overestimates the true learning and performance parameters for below-average students who make many errors. While these students receive more remedial exercises than the above average students, they nevertheless receive less remedial practice than they need and perform worse on the test than expected.

Similarly, when comparing a single BKT model fit to all students (population model) against individualized BKT models, Lee and Brunskill [2012] found that:

> 17% of students would be expected to have a probability of mastery of only 60% or less when the population model would expect the student is at a probability of mastery of 95% or higher

These results suggest that using BKT for mastery learning may result in more higher-performing students reaching mastery than lower-performing students.

Explaining the Problem: Many papers have suggested this problem is due to the lack of individualizing the BKT model to different populations of students. For example, Corbett and Anderson [1995] suggested a way to individualize the parameters of the BKT model in real-time by using the students' error rates when working on the tutoring system. Several other researchers have looked at other methods for fitting BKT parameters that are individualized per student [Lee and Brunskill, 2012, Pardos and Heffernan, 2010, Yudelson et al., 2013]. However, these authors did not explicitly consider the equity implications of this or explicitly quantify how inequitable the models could be. Moreover, while Corbett and Anderson's [1995] individualization algorithm lead to better predicting student posttest scores, they found in future experiments, which used their individualized BKT algorithm, that lower-performing students still tended to achieve lower posttest scores, even though the model predicted nearly all students had mastered the material.

Therefore, lack of individualization may not fully account for why mastery learning may lead to better outcomes for high-performing students.

Contribution: I suggest that inequitable outcomes could result not only from lack of individualization but also from misspecifying the student model. Using simulations, I show that while mastery learning algorithms are substantially more equitable than giving all students the same amount of practice, such algorithms can still be inequitable when they rely on inaccurate models. Model inaccuracies can result both from lack of individualization and from model misspecification. I conclude this section by showing that knowledge tracing with the additive factor model may be more equitable than using BKT, but at the cost of giving extra practice.

7.2 Value Judgments

When discussing ethical concerns such as equity, we must necessarily make value judgments. In this section, we define what we mean when we say a knowledge tracing algorithm is more or less equitable. Although we believe our notion of equity is sensible, we do not mean to convince the reader that our definition of equity is the correct definition or the only definition. Rather, we hope that this work generates discussion around when a learning analytics algorithm should or should not be considered equitable.

In this paper, we assume that an equitable outcome is when students from different populations (e.g., students that have different needs) reach the same level of knowledge after receiving instruction. In what follows, we will typically assume there are only two types of students: either slow learners and fast learners or low prior ability learners and high prior ability learners. The speed of learning and prior ability could be thought of as proxies for separating students who are more or less disadvantaged. For example, low prior ability learners could correspond to students from low socio-economic backgrounds who have had limited access to good instruction or have come into a course with existing gaps in prior knowledge, or slow learners could be coming from a particular demographic background that is more likely to face a stereotype threat or fixed mindset in a particular domain.

Notice that in our notion of equity, the only thing that matters is how much students learned, not how long it took them to reach their level of knowledge. Therefore, we assume it is equitable if slow learners take longer to reach the same level of knowledge as fast learners; it may even be unavoidable. However, if all we cared about was equity, then we could simply give all students inordinate amounts of instruction, but of course that would not be ideal. Therefore, we would ideally want knowledge tracing algorithms that are equitable while minimizing the amount of time wasted on having students do extra practice opportunities. Moreover, a secondary equity concern could be to have slow learners and fast learners have equal amounts of superfluous practice opportunities. Thus, we will also compare algorithms on how much extra practice they give students, but that is not the primary focus of this paper.

	BKT-Slow	BKT-Fast	BKT-Mixed
$P(L_0)$	0.0	0.0	0.071
$P(T)$	0.05	0.3	0.096
$P(G)$	0.2	0.2	0.209
$P(S)$	0.2	0.2	0.203

Table 7.1: BKT Models used in simulations in Section 7.3

7.3 Case 1: Lack of Individualization

We first demonstrate what happens when the BKT mastery learning algorithm uses a single model that is fit to data coming from a mix of slow learners and fast learners. In this case, we are assuming that we are modeling how students learn accurately, except that we incorrectly assume all learners have the same model parameters. This is a weak form of model misspecification, where the misspecification lies in not individualizing the model to different sub-groups of students. As before, we do not mean to actually suggest that learners learn according to BKT or that there are only two types of students. Rather, the goal is to demonstrate how mastery learning could be inequitable *even if* the assumptions of BKT were accurate, but our model is not individualized.

The BKT models we used are shown in Table 7.1. Notice that the two models, BKT-Slow and BKT-Fast, only differ in the $P(T)$ parameter, which we will refer to as the learning rate. Table 7.1 also shows BKT-Mixed, the model that was fit to 200 simulated students from a population of students who were equally likely to come from BKT-Slow and BKT-Fast and who received 20 practice opportunities each. Notice that the key difference between BKT-Mixed and the other models is that it has an intermediate learning rate, since the best fitting model is trying to average over the different rates of learning that were present in the data.

To assess whether the BKT-Mixed algorithm would behave equitably, we ran simulations where BKT-Mixed was used to implement mastery learning for both slow and fast learners. As mentioned before, we are primarily interested in comparing the percentage of slow learners and fast learners who did not learn the skill (even though the algorithm believes they did with 95% certainty), but as a secondary concern, we are also interested in the average amount of extra practice for students who did learn the skill. We first compare the BKT-Mixed algorithm to non-adaptive instructional policies that give a fixed amount of practice to all students. Figure 7.1 shows that for non-adaptive instructional policies, there is a tradeoff between the equity gap between slow and fast learners (i.e., the % of fast learners - % of slow learners that learn the skill) and the amount of extra practice given. However, BKT-Mixed achieves a much better balance between equity and extra practice than the non-adaptive policies. In particular, to achieve the same gap between slow and fast learners for a non-adaptive policy, one would have to give around 46 extra practice opportunities to students on average.

However, Figure 7.1 also confirms that the BKT-Mixed algorithm *is* inequitable, even if much less than non-adaptive algorithms. As shown in Table 7.2, the BKT-Mixed algorithm leads to more slow students (5.5%) not learning the skill than fast students (0.3%), while giving roughly

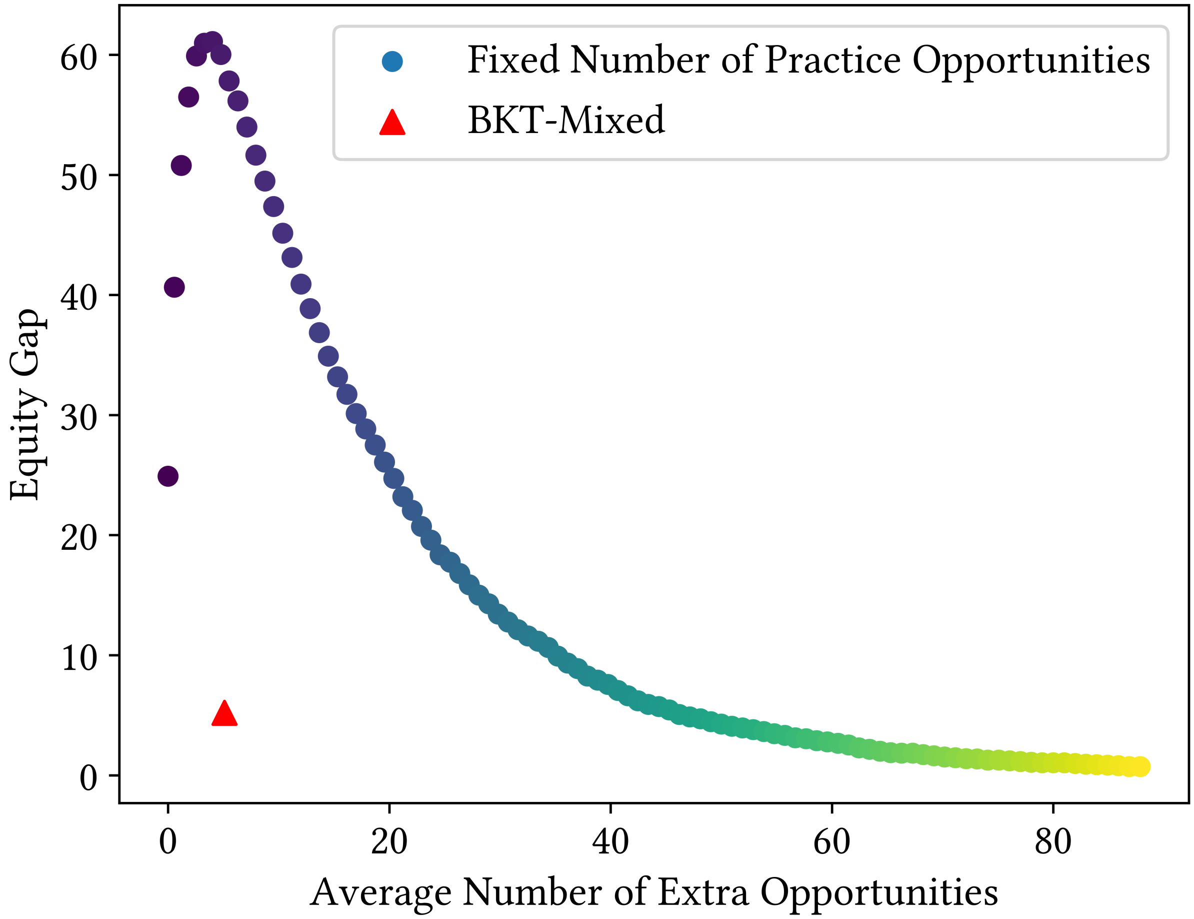


Figure 7.1: Comparison of mastery learning using BKT-Mixed with non-adaptive policies that give a fixed number of practice opportunities (between 1 and 100) to all students. The equity gap refers to the % of fast learners - % of slow learners that learn the skill. The x-axis shows the average amount of extra practice for students that learned the skill.

	Slow Learners			Fast Learners		
	% Not Learned	Avg Extra	Avg % Extra	% Not Learned	Avg Extra	Avg % Extra
BKT-Mixed	5.5%	5.0	89%	0.3%	5.1	269%
True Models	3.0%	5.6	96.8%	2.0%	3.3	180%
3-CCR	10.3%	4.4	82%	0.9%	4.4	230%
5-CCR	0.4%	9.9	159%	0.02%	9.8	497%
AFM	0.2%	19.4	193%	0.02%	9.7	472%

Table 7.2: Effects of simulating various mastery algorithms on simulated BKT learners with different learning rates. Each simulation consisted of either 5000 BKT-Slow learners or 5000 BKT-Fast learners who were given practice opportunities until they reached mastery according to the knowledge tracing algorithm for that row. For each type of learner, the first column shows the percentage of students that did not learn the skill, the second column shows the average number of superfluous practice opportunities for students who learned the skill, and the third column shows the average percentage increase in number of practice opportunities beyond those that the student needed to learn the skill.

the same amount of extra practice for both slow and fast learners. This problem could be resolved in one of two ways. First, individualized BKT models could be fit per student or per cluster of students to account for the different learning rates. Second, the mastery threshold could be increased, at the expense of giving learners more unnecessary practice. The second row of Table 7.2 shows that if we had used the true model (BKT-Slow for slow learners and BKT-Fast for fast learners), the gap between slow and fast learners diminishes[1]. Notice that using these individualized models, the outcomes for fast learners are seemingly worse than when using BKT-Mixed, given our stance that not learning the skill is worse than receiving additional practice. This is simply because we are using a threshold of 95%, which states that we are comfortable with 5% of students not reaching mastery; if we would rather have more students reach mastery at the expense of students receiving extra practice we could simply increase the mastery threshold. In some sense, the mastery threshold represents the degree to which we are willing to tradeoff between the percentage of students that learn the skill and the amount of extra practice.

In addition to the BKT algorithm, the third row of Table 7.2 shows that the the 3-CCR heuristic, which has been used in the ASSISTments system [Beck and Gong, 2013], is also inequitable. In fact, the achievement gap is higher than for BKT (10.3% of slow learners do not learn the skill vs. 0.9% of fast learners). Notice that although N-CCR does not explicitly use a model, it still makes the assumption that all learners are the same, which is the root of the problem. Again, this could be fixed either by choosing an individualized value of N for each student or each type of learner (e.g., slow and fast), or by uniformly increasing the value of N, which is analogous to increasing the mastery threshold for BKT. In this case, $N = 5$ is sufficient, as seen in the fourth row of Table 7.2. However, it may be unclear a priori how large N needs to be,

[1] There are still 1% more slow learners who do not reach mastery than fast learners, but this difference is not systematic; for some learning rates, slow learners have a slightly higher percentage of learning the skill instead.

	BKT-Slow$_{\text{AFM}}$	BKT-Fast$_{\text{AFM}}$	BKT-Mixed$_{\text{AFM}}$	BKT-Low$_{\text{AFM}}$	BKT-High$_{\text{AFM}}$
$P(L_0)$	0.123	0.055	0.065	0.001	0.001
$P(T)$	0.021	0.043	0.022	0.114	0.173
$P(G)$	0.123	0.151	0.152	0.125	0.509
$P(S)$	0.448	0.435	0.41	0.612	0.205

Table 7.3: BKT models used in simulations where AFM is the true model that describes student learning

and if N is too large, students may be doing much more practice than necessary. For example, Khan Academy found that even with $N = 10$, students who took more problems to get 10 correct in a row (presumably slower learners) were more likely to get the next question incorrect than students that immediately got 10 in a row correct [Hu, 2011].

7.4 Case 2: Model Misspecification

In the previous section, we saw that if there are students who learn at different rates, and we do not account for that in modeling student learning, then knowledge tracing could lead to widening the achievement gap. In this section, we look at the more fundamental problem where the model of student learning is entirely misspecified. In particular, we show that if learning is described by the additive factor model, then using BKT can lead to inequitable outcomes, *even if* we fit different parameters for different types of learners. We begin with an analysis of slow and fast learners, and then move on to low and high prior ability learners.

7.4.1 Different Learning Speeds

We defined two AFM models with the only difference being that one model had a learning rate (γ) of 0.05 (AFM-Slow) and the other model had a learning rate of 0.1 (AFM-Fast). For both models, the student abilities (θ) were sampled from a $\textsc{normal}(-2, 1)$ distribution. We then fit a BKT model to 200 simulated students from a population of students who were equally likely to come from AFM-Slow and AFM-Fast and who received 20 practice opportunities each. We refer to this model as BKT-Mixed$_{\text{AFM}}$. We also fit a BKT model to a population of only 200 AFM-Slow learners (BKT-Slow$_{\text{AFM}}$) and a BKT model to a population of only 200 AFM-Fast learners (BKT-Fast$_{\text{AFM}}$). These models are shown in Table 7.3. Notice that the key difference between the models fit to slow and fast learners is the learning rate, which makes sense as that is the only difference between the AFM models; interestingly, the mixed model has a learning rate that is almost identical to that of the slow model, but the rest of the parameters have values closer to those of the fast model.

Because AFM is an incremental model of learning, it no longer makes sense to evaluate whether or not students learned the skill or how many extra practice opportunities they received. Instead, we are interested in the probability of students answering the next item correctly at the point

	Slow Learners	Fast Learners
BKT-Mixed$_{\text{AFM}}$	0.49	0.60
BKT-Slow$_{\text{AFM}}$	0.45	-
BKT-Fast$_{\text{AFM}}$	-	0.56
3-CCR	0.47	0.58
5-CCR	0.67	0.76
AFM	0.955	0.962

Table 7.4: Effects of simulating various mastery learning algorithms on simulated AFM learners with different learning rates. Each simulation consisted of either 500 AFM-Slow learners or 500 AFM-Fast learners who were given practice opportunities until they reached mastery according to the knowledge tracing algorithm for that row. Both columns show the average probability of answering an item correctly when the knowledge tracing algorithm declares mastery.

where the knowledge tracing algorithm declares mastery. An equitable outcome would be to have an identical probability of correct answers for fast and slow learners. Table 7.4 shows the results of simulations of knowledge tracing with the three fitted BKT models on AFM-Slow students and AFM-Fast students. What we find is that there is a 0.11 difference in the probability of correctness at mastery for slow and fast learners. Moreover, this difference does not go away even when we individualize the BKT models (i.e., use the BKT models specifically fit to slow and fast learners). A similar gap exists when using the 3-CCR algorithm, and in this case the gap barely decreases when using the 5-CCR algorithm. Thus, when the model used in knowledge tracing is incorrect, even individualization of the model may not be sufficient to eliminate the achievement gap.

To demonstrate that the problem in this case is indeed model misspecification, we also used AFM to perform knowledge tracing. Recall that AFM assumes each student has an individualized ability parameter (θ). Assuming we do not have knowledge of this ability ahead of time, it makes sense to estimate it in an online fashion. Moreover, because we also want to distinguish between slow and fast learners, it would be ideal if we could learn the learning rate (γ) online. Thus, to use AFM for knowledge tracing, we fit the logistic regression model at each time step using all the practice opportunities for that student so far as data. Additionally, we decided that the AFM-based algorithm would declare mastery when the estimated probability of correctness reached 0.95. To ensure that the student had actually reached the threshold (rather than the parameters being spuriously estimated), we used the minimum estimated probability of correctness over the last three practice opportunities. As shown in Table 7.4, this algorithm was equitable in that slow and fast learners both reached the desired level of mastery (0.95).

Since this online AFM-based knowledge tracing can estimate the learning rate on-the-fly, we wanted to see if it would be equitable when used on BKT learners. To test the AFM algorithm on BKT learners, we used the same procedure as described above, except that we set the mastery threshold to 0.8, as that is the highest accuracy that students can reach under a BKT model with a slip probability of 0.2. The last row of Table 7.2 shows that using this algorithm leads to mastery for all learners, at the expense of giving more practice opportunities than needed, especially for slow learners. While the amount of extra practice for slow learners is not ideal, this suggests that perhaps the AFM algorithm is more equitable, even under model misspecification. However, more work is needed in assessing the equitablity of AFM-based knowledge tracing assuming different types of model misspecification.

7.4.2 Different Prior Ability Levels

Everything we have discussed so far was assuming students differ in terms of how fast they learn, but in actuality student differences may be more salient in terms of their prior knowledge and ability rather than learning rate. Recall that in the AFM-Slow and AFM-Fast models used above, student ability was actually allowed to vary ($\theta \sim \mathcal{N}(-2, 1)$). Therefore, even though we were supposing there were only two types of learners, in reality each learner had their own level of prior ability. Figure 7.2 shows that for AFM-Slow learners, the probability that a learner answers correctly when BKT-Slow declares mastery varies drastically for learners with different prior abilities. That is, learners with above-average ability tend to reach much higher levels of mastery than students with average or below-average ability. On the other hand, the same plot shows that when we use the AFM algorithm for mastery learning, almost all simulated students reached higher levels of mastery than *all* of the simulated students interacting with the misspecified BKT-Slow algorithm. However, we also find that AFM-learners with higher ability sometimes end up with lower accuracy at mastery than initially lower ability students. This is the first time when an algorithm appears inequitable with worse outcomes for higher-performing students. The reason for this inequitable outcome is different from the others; since high performing students answer questions correctly with high probability, it is more likely that the algorithm will encounter a string of consecutive correct responses early on, which would lead the AFM algorithm to think students have reached a higher level of accuracy than they actually have. This could perhaps be corrected by modifying the algorithm to only declare mastery when it can make a prediction that a learner has reached a high accuracy state with high confidence, but we leave this as an interesting direction for future work.

In the analysis above, we did not individualize the BKT models with respect to prior ability. To see if individualization could help address this issue, we assume there are two types of AFM learners as before, but instead of differing in terms of learning rate, these learners only differ in terms of student ability. That is, both types of learners have a learning rate of 0.1, but low-ability learners have a student ability of -2 (AFM-Low) and high ability learners have a student ability of 0 (AFM-High). As before, we fit a BKT model to a population of 200 AFM-Low learners (BKT-Low$_{\text{AFM}}$) and a separate BKT model to a population of 200 AFM-High learners (BKT-HighAFM), with each learner receiving 20 practice opportunities. These fitted BKT models are shown in Table 7.3. Interestingly, we see that the two fitted models have very different (and semantically degenerate) guess and slip parameters. This is because the best fitting 2-state HMMs will be very different for students that start at different initial knowledge states (i.e., different probabilities of answering items correctly). If we use these models to perform mastery learning with 500 simulated AFM learners (either AFM-Low or AFM-High), we find that the

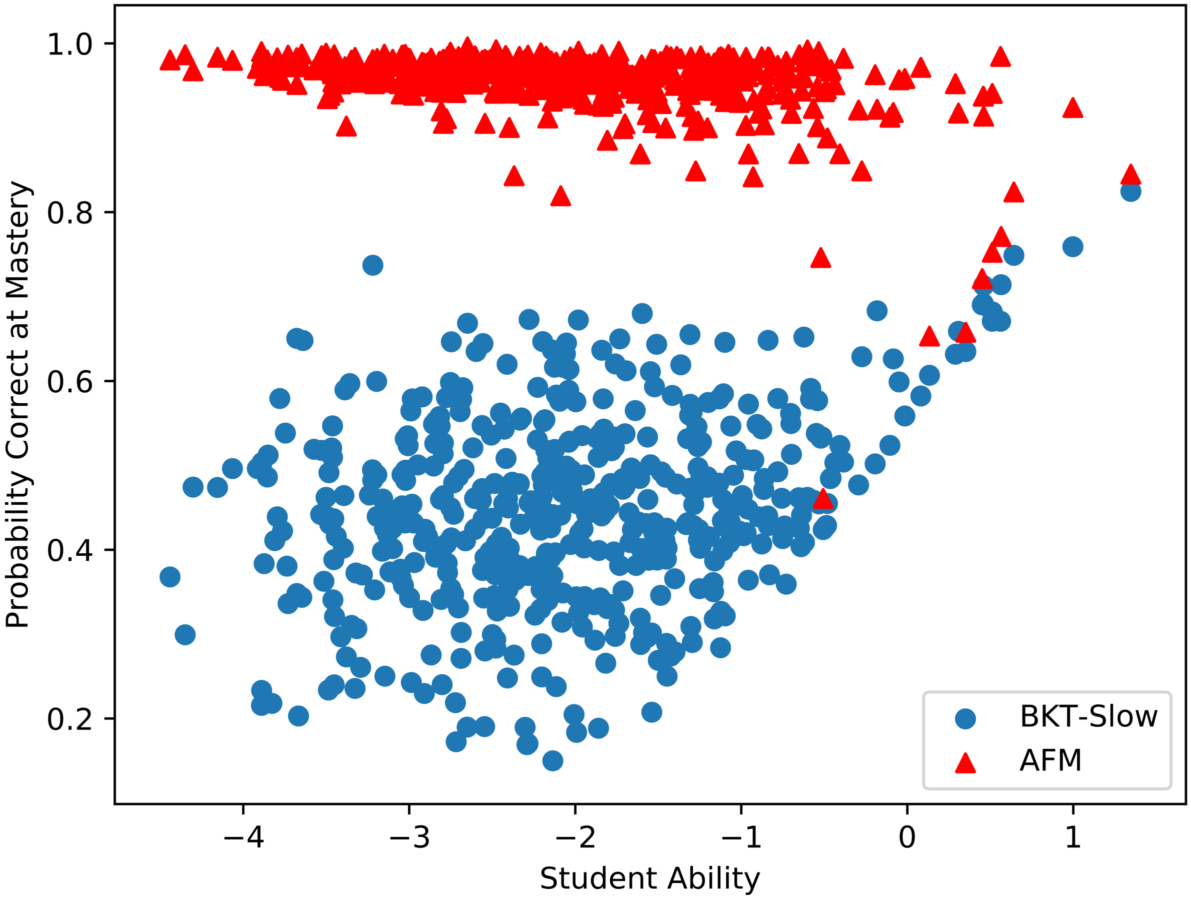


Figure 7.2: The probability of answering an item correctly for AFM-Slow learners when mastery is declared vs. student ability (θ). The blue circles show this for 500 simulations using BKT-Slow as the mastery learning algorithm and the red triangles show 500 simulations using AFM as the mastery learning algorithm.

average probability of correct at mastery is 0.41 for AFM-Low learners and 0.77 for AFM-High learners, indicating a huge equity gap.

7.5 Discussion

By allowing all students to go through curricula at their own pace to ultimately reach mastery, adaptive educational algorithms such as knowledge tracing are implicitly meant to eliminate inequities between different groups of students. While this is a noble goal, we have shown that due to differences between the way we model student learning and how students actually learn, we may sometimes fall short of reaching this goal. Although the degree of inequity in some cases may seem small (e.g., 5% of slow learners are lagged behind), these inequities could cascade as the amount of content increases and as content builds on assumed prior knowledge. However, the observation that using AFM might lead to more equitable outcomes makes us optimistic that there exist approaches to knowledge tracing that are equitable in a variety of settings. However, the AFM algorithm we described has some limitations. It acheives equitable outcomes by being overly conservative (i.e., giving too much extra practice). Moreover, the AFM algorithm may declare mastery too early for students who start out with high ability due to lack of sufficient data to accurately predict these students' knowledge states.

We believe there are several concrete steps that can be taken to design more equitable knowledge tracing algorithms. First, we should consider the equity of knowledge tracing in more realistic settings, such as when students are learning multiple skills sequentially. A proper notion of equity in this setting must consider both the degree to which skills are learned as well as how many skills are learned. More realistic simulations could also consider students coming in with specific gaps in prior knowledge, rather than characterizing student ability on a single dimension. Second, we should evaluate the equitability of different knowledge tracing algorithms under various assumptions about how students learn. This could lead to finding ways to refine our algorithms to make them more equitable in diverse settings. The robust evaluation matrix I introduce in the next chapter provides a tool for accomplishing this. Finally, we should move beyond simulations to see if algorithms that are predicted to be equitable actually live up to that promise when used to teach actual learners, and if not, how we can refine our simulations to develop more equitable algorithms.

Chapter 8

Robust Evaluation Matrix[*]

> We attempt to treat the same problem with several alternative models each with different simplifications but with a common...assumption. Then, if these models, despite their different assumptions, lead to similar results, we have what we can call a robust theorem that is relatively free of the details of the model. Hence, our truth is the intersection of independent lies.
>
> Richard Levins, 1966

In the previous two chapters, we saw that the Bayesian knowledge tracing model, which is commonly used to perform cognitive mastery learning might not fully live up to the promise of giving all students the right amount of practice. We now consider broader instructional policies for sequencing content for students. When using model-based reinforcement learning for instructional sequencing, we assume there is a model that describes how students learn, typically in the form of a Markov decision process or partially observable Markov decision process. Even though these models can be learned from data, someone must specify the state space inherent to these models. For example, in an educational context the state of a student could be a latent knowledge state for each knowledge component (as is used in BKT) or the past *N* responses that the student gave for each knowledge component. The Markov assumption inherent to MDPs and POMDPs claims that the next state of a student only depends on the student's current state. Therefore, if we choose a state representation that cannot accurately capture the complexity of student learning under the Markov assumption, we again suffer from model misspecification. What are the implications of this model misspecification on the learned content selection policies? In particular, we will consider the impact of model misspecification on trying to evaluate different instructional policies for purposes of choosing a policy to deploy in practice using prior data.

We show that relying on a single model to estimate the impact of an instructional policy can lead to misinformed decisions for similar reasons to how using BKT could lead to misguided estimates about when students reach mastery. On the other hand, I briefly discuss how statis-

[*] This chapter was adapted from Doroudi et al. [2017a].

tical estimation techniques that do not rely on student models, like importance sampling, are inherently high variance and hence have limited applicability to educational domains when we want to make many sequential decisions. To mitigate these two extremes, we present the robust evaluation matrix (REM) method for estimating the potential impact of a new way of teaching (focusing on sequencing strategies) in advance of running an experiment. REM seeks to make our predictions more robust to model misspecification by not assuming students learn according to any particular model. Instead, REM evaluates how well instructional policies perform on multiple student models, and can benefit from using a diversity of models, which in theory could be motivated by various theories of learning. The various techniques to evaluating and selecting instructional policies that we discuss in this chapter are shown in Figure 8.1.

The idea of using multiple models to mitigate the bias-variance tradeoff is similar to work on model ensembles in the machine learning literature. Ensemble learning techniques, such as bagging, boosting, and stacking, utilize multiple models to reduce the bias and/or variance of the individual models when making predictions. For example, stacking or stacked generalization can take multiple models as input and then use a meta-algorithm (such as linear regression) to assign a weight to each algorithm (for example, as linear regression) to make a final prediction that could correct for biases in the input models [Breiman, 1996, Wolpert, 1992]. In reinforcement learning, model ensembles have also been used to make policies that are more robust to the underlying transition dynamics [Rajeswaran et al., 2016]. Our work has a similar flavor, but here we do not make a single prediction by combining multiple models; rather each model makes its own prediction. A robust instructional policy is one that does well according to a variety of models. It is ultimately up to a human (e.g., researcher or instructional designer) to determine when an instructional policy is robust enough to be used with real students.

We present two case studies to show how REM can be used in practice. In the first, we demonstrate that our method can help correctly predict when a new policy would not be effective in improving student learning while standard model-based evaluation predicts otherwise. We then attempt to use REM to find an instructional policy that is robust to how students learn. While we ultimately find that the policy that REM predicted would be good was actually not better than a baseline policy, this process gave us further insights into the limitations of the models we used in REM. In the second case study, we retrospectively analyze work by Rafferty et al. [2015b] to show that REM could be used to detect policies that will do better than baselines in a concept learning domain when deployed on actual students, and again can detect cases where policies are likely to be ineffective in the real world.

Before describing REM, we discuss prior work in trying to automatically choose instructional policies using prior data and the limitations of existing approaches.

8.1 Off-Policy Policy Estimation and Selection

In this section, we investigate the impact of model misspecification on the related problems of off-policy policy estimation and off-policy policy selection: the setting where we have access to

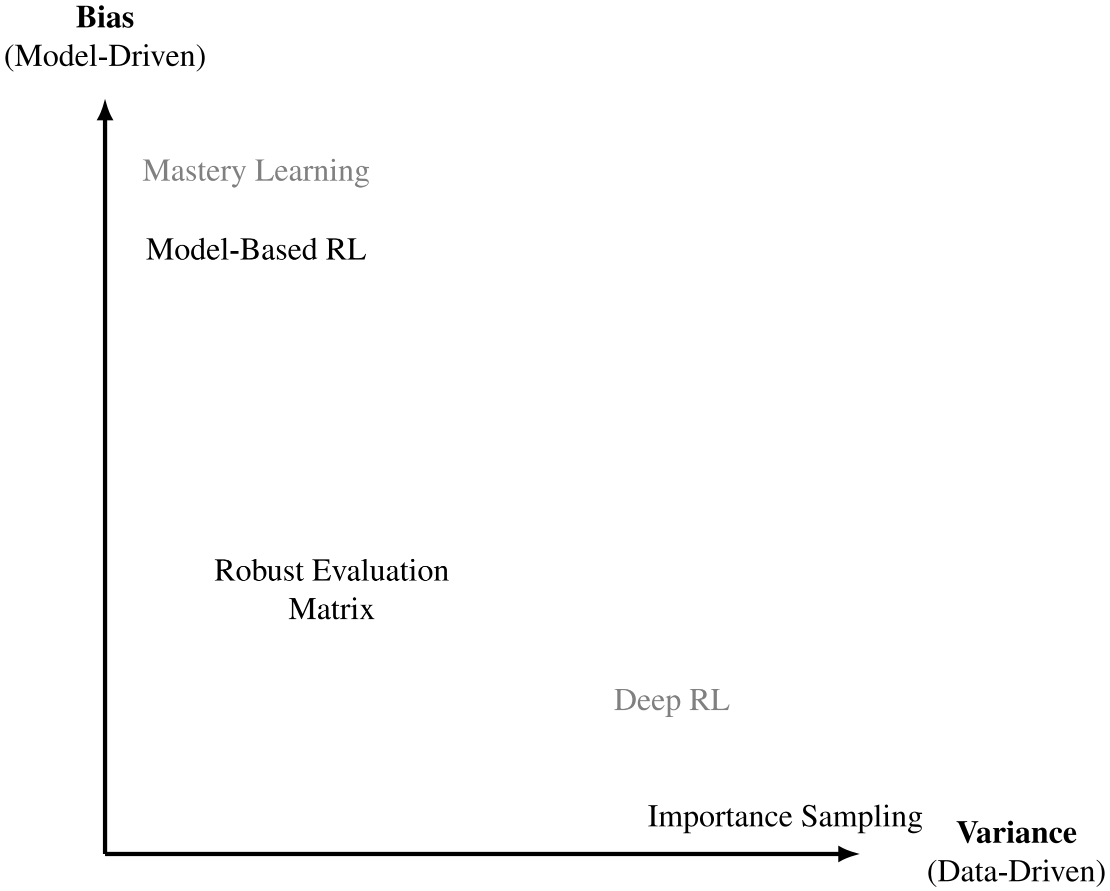


Figure 8.1: The techniques to evaluating and selecting instructional policies discussed in this chapter (shown in black) positioned on the bias-variance tradeoff

prior data collected using some policy, and we want to use that data to make inferences about one or more *other* (instructional) policies. Off-policy policy estimation can be used to estimate the performance of a new instructional policy without (or in advance of) running an experiment. Such counterfactual reasoning is important not just in education, but in a wide swath of other areas including economics, healthcare, and consumer modeling [Thomas et al., 2015, Zhou and Brunskill, 2016]. Off-policy policy estimation is often a critical part of off-policy policy selection: determining which policy from among a set of candidate policies would have the highest expected performance if deployed in the future. We are primarily interested in the problem of off-policy policy selection, as it can have practical implications with respect to what we do in practice. We consider the problem of off-policy policy estimation in so far as it helps us achieve the former. While the two have been tightly coupled in the literature, we show that this need not be the case; in the next section, we will present a method that does not necessarily give us reliable estimates of the performance of instructional policies but could still be used to compare instructional policies. We now discuss approaches to doing off-policy policy estimation and selection, including how this problem has been tackled in education settings.

8.2 Model-Based Evaluation

Ideally, we would like to evaluate the efficacy of an instructional policy before committing to using it on actual students. Running experiments to test the efficacy of various instructional policies on actual students can be costly and time consuming, and if the policies aren't helping students, then we could be wasting student time that could be spent on more valuable learning experiences. Instead, it would be useful if we could evaluate how good a content selection policy might be before testing it on students. This problem is called **off-policy policy evaluation**. A standard way of doing this is **model-based evaluation**, i.e. to simulate the instructional policy on a particular model of student learning to evaluate how good the instructional policy is under the assumption that that particular student model is correct. Often times, an instructional policy is derived to optimize student learning assuming a particular model of student learning, and so we simulate the instructional policy on the model that was used to derive it. We call this **direct model-based evaluation**.

This approach has been used to compare and select among different student models and their optimal instructional policies. Chi et al. [2011] used this approach to select an instructional policy, by comparing different student models represented as Markov decision processes with different student features and the resulting instructional policy that yielded the best expected performance for a given model. Similarly, Rowe et al. [2014] estimated the predicted performance of instructional policies that were designed to maximize performance under particular student models and compared them to some hand designed baseline policies and a random policy by evaluating these instructional policies under the same student models. Unsurprisingly, the policy that was computed to have the best predicted performance for a given student model was also estimated to to out-perform the baseline instructional policies under that same model.

This approach is quite appealing, as it is more directly getting at what we often care about: estimating the performance of instructional policies in order to select an instructional policy with the best expected performance. Unfortunately, due to model misspecification, evaluating a policy assuming the student model it was derived under is correct will generally not provide an accurate estimate of the value of a policy if it were to be used with real students. Comparing the estimated performance of instructional policies when each policy is evaluated using a different simulated student model can therefore yield misleading conclusions. Indeed, Mandel et al. [2014a] have shown that *even if* the real world can be accurately modeled as a complex Markov decision process, it is possible that the optimal policy for an alternate statistical model that is incorrect might have a higher estimated performance than the optimal policy of the true MDP, even with an infinite amount of data.

Indeed, the limitations of evaluating the performance of a policy with the student model used to derive the policy has been observed previously. In simulation, Rowe et al. [2014] estimated a new instructional policy would have a performance of 25.4 in contrast to a random policy that was estimated to have a performance of 3.6, where performance was measured as a function of students' normalized learning gains[1] beyond the median student and the performance of both policies was simulated with the student model used to derive the new instructional policy. In contrast, in an experiment with real students, there was no significant difference between the performance of students taught by the two policies [Rowe and Lester, 2015]. While there are many factors in any experiment with real students, estimating performance using the assumed student model may particularly lead to overly optimistic estimates of the resulting performance. In the next section, we provide another example of how direct-model based led to over-predicting the value of policies in our experiment.

8.3 Importance Sampling

Using prior data to obtain an estimator of a content selection policy's performance in advance of deploying the new policy that is not biased by assuming a particular statistical student model could seem rather difficult. However, there does exist an elegant solution: importance sampling, an approach that does not require building a student model, but rather re-weighs past data to compute an estimate of the performance of a new policy [Precup, 2000]. Importance sampling is statistically consistent and unbiased. In prior work, Mandel et al. [2014a] used importance sampling to find an instructional policy in an educational game that significantly outperformed a random policy and even an expert-designed instructional policy. Unfortunately, importance sampling tends to yield highly variable estimates of a new policy's performance when evaluating instructional policies that are used for many sequential decisions, such as students interacting with a tutoring system across many activities. Intuitively this issue arises when a new policy is quite different from a previous policy, and so the old data consists of quite different student trajectories (sequences of pedagogical activities given and student responses) than what would be expected to be observed under a new policy. Mathematically, this is because importance sampling yields unbiased but high variance estimates, unlike direct-model based evaluation which can yield very biased estimates (due to model misspecification) with potentially low variance (when we have enough data).

It is true that with more data, the variance of the importance sampling estimator will decrease, so one may assume this should be the method of choice for learning at scale, but this is not the case when one has to make a large number of sequential decisions. For example, consider some educational software that presents 20 activities to students and only needs to choose between one of two options at any given time (for example, whether to give the student a worked example or a problem-solving exercise). Suppose we have collected existing data from a policy that randomly chose each option for each of the 20 decisions and want to use this for off-policy policy estimation. If we want to evaluate a deterministic instructional policy (i.e., a policy with no randomness), then only one out of every 2^{20} (over one million) students would encounter a trajectory that matches the policy of interest, which means we need millions of students to get a decent estimate of the policy. If the software were to make 50 decisions, then we would need

[1] The normalized learning gain for a student is the difference between the posttest score and pretest score of the student divided by the maximum possible difference.

over $2^{50} \approx 10^{15}$ students!

Finding a statistical estimator that offers the best of both approaches (model-based evaluation and importance sampling estimators) is an active area of research in the reinforcement learning community [Dudík et al., 2011, Jiang and Li, 2015, Thomas and Brunskill, 2016]. Yet it remains a challenge whenever the (instructional) policies are to make a large number of sequential decisions, as highlighted above. Moreover, I have recently shown that if we want to evaluate two policies using importance sampling in order to pick the best policy, importance sampling can sometimes favor the worse of the two policies more often than not [Doroudi et al., 2017b]. Similarly, if we have a set of candidate policies and we want to find out which one is best, importance sampling can tend to favor a sub-optimal policy. This problem is essentially due to the fact that the importance sampling estimator is an asymmetric, high-variance distribution. We have shown that this problem can naturally arise whenever we have trajectories of varying length, such as when students do varying numbers of problems on an intelligent tutoring system or educational game, which is typically the case.

Thus, while direct model-based evaluation is biased due to model misspecification, importance sampling has high variance, which limits its practicality, even though it is unbiased. These approaches are, in some sense, on two extremes of the bias-variance tradoeff in machine learning. In what follows, we consider a way to navigate the bias-variance tradeoff by leveraging multiple (biased) models of student learning.

8.4 Robust Evaluation Matrix

Ideally we want a method for off-policy policy estimation that combines the statistical efficiency of (student) model based estimators with the agnosticism of importance sampling techniques which allows them to be robust to the choice of student model used to derive a particular policy. As we previously argued, this is important even given an enormous amount of data. One potential avenue is to focus on designing better student models, a key effort in the educational data mining and artificial intelligence in education communities. However, since these model classes will still likely be approximate models of student learning, we propose an alternative approach that may not enable us to achieve accurate estimates, but can still help inform comparisons among different policies: using many models we expect to be wrong, rather than using one model we hope to be right.

Our robust evaluation matrix (REM) is a tool for more conservatively evaluating the potential performance of a new policy in relation to other policies during off-policy policy selection. As shown in Algorithm 1, the simple idea is to estimate the performance of different instructional policies by simulating them using multiple plausible student models whose model parameters were fit using previously collected data. The rows of the matrix are different student models and the columns of the matrix are the various policies one wants to estimate the performance of. An entry in the matrix represents the expected performance of a particular instructional policy when simulated under a particular student model. As the student model simulators have parameters that are fit based on the previously collected data, they will often represent reasonable possible ways of modeling the dynamics of student learning. If we restrict our comparison to models with similar predictive accuracy (e.g., as evaluated using cross validation or a test set constructed from the available data), it is unclear which model is better, but the REM method can be used to assess trends in performance across policies that are consistent across multiple possible ways that students may learn in the real environment (e.g., Bayesian Knowledge Tracing, Performance Factors Analysis, Deep Knowledge Tracing etc.).

Simulating the potential performance of instructional policies under multiple student models to inform off-policy policy selection has been previously underexplored. There has been some prior work that analyzes the interaction of student models and instructional policies (that may have been derived with a particular student model) [Clement et al., 2016, González-Brenes and Huang, 2015, Lee and Brunskill, 2012, Rafferty et al., 2015a, Rollinson and Brunskill, 2015], but such work has often been done to understand the general differences between policies run on various models, rather than as a tool to inform whether a new policy may offer benefits over previous ones before conducting experiments or embedding a policy in a tutoring system. One exception is work by Clement et al. [2016], where they investigate the case where the knowledge graphs (i.e., prerequisite relations between knowledge components) used to learn models used to compute instructional policies are not the same as the ones underlying student learning. The authors found that a particular model that does not have parameters fine-tuned to the knowledge graph performs best when there is a mismatch in the policy's representation of knowledge graph and true knowledge graphs of students. Their work differs from our current paper in that the authors only consider robustness of policy's of varying complexity in light of the knowledge graph changing but do not consider student models that differ more wildly and the authors do not present a general method for off-policy policy estimation or selection. Moreover, they only presented results from simulations with hand-crafted parameters rather than models and policies fit to real data. Nonetheless, we can consider this work as an example of REM being used in the past to inform policy selection. The most closely related work is by Rafferty et al. [2015a], which analyzed the potential performance of various instructional policies derived from different models of student concept learning under various student concept learning models that were fit from a previously collected dataset. However, unlike our current paper, they presented this idea primarily to understand the interaction between the policies and the models of student learning (e.g. could a policy assuming a very simple model of student learning still do well if the real student exhibits much more complicated student learning), rather than as a generic tool for off-policy policy estimation and selection. In the next section, we reinterpret their results as a positive use case of REM. Moreover, while Rafferty et al. [2015a] consider simulating policies only on models of student learning that were used to derive some of the policies, REM could simulate policies on other models of student learning, even if one does not derive any policies from those student models. We present one example of this in the next section.

REM can be used in several ways. If one or more student models in the matrix suggest that a new policy is no better or even worse than other (baseline) policies, then it would suggest a new policy may not yield a significant improvement in learning outcomes. On the other hand, if the student models agree that one policy appears to be better than others (and these student models

Algorithm 1: Pseudocode for algorithm to construct robust evaluation matrix

```
Input: Set of student models $m = 1 \dots M$ and instructional policies $p = 1 \dots P$
REM ← $m \times p$ matrix
for model $m = 1 \dots M$ do
    for policy $p$ = 1... $P$ do
        if student model $m$ compatible with instructional policy $p$ then
            mean, stddev ← Estimate performance of instructional policy $p$ on student model $m$
                // For example by simulating many times
            REM[$m$][$p$] ← mean, stddev
return REM
```

are indeed quite different from each other[2]), then it should increase our confidence that the policy will actually out-perform the other policies. Recall that we are interested in the joint problems of off-policy policy estimation and off-policy policy selection. We propose that REM can help with addressing the second problem, even though it does not necessarily help us with the first. That is, if we find a policy that robustly does better than another policy according to various student models, then we may decide to choose to implement that policy in practice; however, if different student models have very different predictions as to how well the new policy will perform, then we may not have a good estimate of its performance a priori. But having an estimate of a policy we are confident will do well a priori may not be necessary if we are planning on testing it on actual students anyways. This makes REM differ from off-policy policy selection techniques in the existing literature, which aim to use imperfect methods of policy estimation as a way to do policy selection. Rather, REM aims to help the researcher make decisions about what policy to select without directly trying to get a good estimate of a policy's performance. Notice that REM does not decide for us when to use a particular content selection policy in practice; that is REM is not a black box reinforcement learning algorithm, it is a human-in-the-loop algorithm that relies on the user's understanding of the models and policies to make the decision if there is sufficient confidence to test an instructional policy on students. Furthermore, REM may give insights to the designer if there seem to be limitations to the models or policies being used.

8.5 Case Studies

We now present two case studies to ground the discussion and illustrate how REM can inform what instructional policies may yield improved performance, given prior data. The first is an experimental study we ran in which we used old data to derive a new policy we estimated to be better than a standard baseline, but which yielded equivalent performance in a subsequent student study. Our post hoc analysis suggests we could have predicted this result by using a

[2] The difference in student models could be based difference in theory, for example a Bayesian Knowledge Tracing model and a Deep Knowledge Tracing model make rather different assumptions about student learning—or based on empirically observing that simulating the same instructional policy on two different models results in reasonably different trajectories quantified in some way.

REM analysis. In the second case study, we will look at the results of a paper by Rafferty et al. where they perform an analogue of REM to better understand how various instructional policies might perform under different student models [Rafferty et al., 2015a]. Although their paper did not suggest using such a method for off-policy policy selection, we show two examples of how it could have been used both to predict that several policies were likely to do well when tested on real students and to predict that another policy may perform poorly (a result that would not have been predicted if a policy's performance was only estimated assuming that the student model used to derive the policy was in fact how students truly learn).

8.5.1 Case Study 1: Fractions Tutor Experiment

To ground the discussion, we now present a case study of an experiment we ran on our fractions intelligent tutoring system. We will discuss how we used old data to derive two new adaptive content selection policies we estimated to be better than a standard baseline, but which yielded equivalent performance in a subsequent student study. We then show how our post hoc analysis suggests we could have predicted this result by using REM. We then used REM to inform the choice of a new adaptive content selection policy for a second experiment, and although the second experiment was also not successful, we discuss additional factors we should consider in REM and the insights we gained from doing this analysis.

We ran an experiment to test five instructional policies in an intelligent tutoring system (ITS) designed to teach fractions to elementary school students [Doroudi et al., 2015, Rau et al., 2013]. There were two main goals to the experiment:

1. to test whether adaptive problem selection based on an individual student's knowledge state makes a difference (in terms of improving student learning), and
2. to test whether supporting a variety of activity types in an ITS leads to more robust learning.

Additionally, we were interested in testing whether we could improve upon the traditional form of adaptive instruction used in ITSs: cognitive mastery learning using Bayesian Knowledge Tracing (BKT). Namely, we were interested in testing whether reasoning about (prerequisite) relationships between skills when deciding what problem to give a student to solve improves student learning beyond simply giving problems until a student masters each skill independently. We therefore developed a new student model that treats the correctness on the last two steps of each skill as the state of a student's knowledge of that skill, and then predicts the student's next state of a skill based on the student's knowledge of that skill as well as prerequisite skills. Prerequisite skills were identified using the G-SCOPE algorithm [Hallak et al., 2015]. Our models used a skill model that was inferred using the weighted Chinese restaurant process technique developed by Lindsey et al. [2014a], which was seeded with a hand-crafted skill model. Model parameters were fit given access to data that was previously collected using a semi-random instructional policy to teach over 1,000 students, who used the tutor for four to six days, with most students completing between 20 and 100 problems out of a potential set of 156 problems. Student learning was assessed using identical pretests and posttests composed of 16 questions.

	Instructional Policies				
	Baseline 1	Baseline 2	BKT-MP	AP-1	AP-2
Direct Model-Based Evaluation Results	5.87 ± 0.90	6.10 ± 0.97	7.03 ± 1.00	7.85 ± 0.98	9.10 ± 0.80
Actual Experimental Results	5.52 ± 2.61	5.14 ± 3.22	5.46 ± 3.0	5.57 ± 3.27	4.93 ± 1.8

Table 8.1: The first row shows the estimated expected performance of a student when taught under each policy, assuming either the student model used to derive the policy, or, in the case of the non-adaptive policies, using the estimated G-SCOPE student model. The second row shows the results of our actual experiment. Note that the posttest was out of sixteen points.

We iterated over multiple potential adaptive instructional policies, seeking to identify an instructional policy that we estimated would yield improved performance over both strong baseline non-adaptive instructional policies, and equal or better performance to a state-of-the-art policy based on a mastery learning instructional policy.

Since each student completed many problems using the tutor, typically more than 20, importance sampling techniques for estimating the student learning outcomes under an alternate instructional policy (that adaptively sequenced activities in a different way) were infeasible (see example in Section 8.3). Instead, we relied on simulating a policy's performance based on a student learning model. We chose adaptive policies that we estimated would yield a significant improvement over the non-adaptive baselines. This lead us to choose the following adaptive content selection policies for use in a future experiment, policies that we believed had a good chance of yielding a significant improvement,

- Adaptive Policy 1 (**AP-1**): greedily maximize the number of skills that students learn with each problem assuming the fit G-SCOPE model.
- Adaptive Policy 2 (**AP-2**): Selects problems to myopically maximize the student's posttest score under a fit G-SCOPE student model.

These were to compared to the following baselines

- Baseline 1: Instructional policy that selects standard (induction and refinement) problems, in a reasonable non-adaptive order, based on spiralling through the curriculum.
- Baseline 2: Instructional policy that selects among a diverse set of problem types, in a reasonable non-adaptive order, based on spiralling through the curriculum.
- BKT Mastery Policy (**BKT-MP**): This is a cognitive mastery learning policy used with a BKT model, which always selects the problem where students are expected to learn the most skills.

Row 1 of Table 8.1 shows the estimated performance of the above policies, where each adaptive policy was simulated using the student model used to derive the policy. Since the first two policies are non-adaptive, they were not derived using a student model. We used the G-SCOPE student model to simulate the performance of these baseline non-adaptive policies. All evaluations assumed each (simulated) student completed 40 problems, and we repeated this process

		Instructional Policies				
		Baseline 1	Baseline 2	BKT-MP	AP-1	AP-2
Student Models	G-SCOPE Model	5.87 ± 0.90	6.10 ± 0.97	N/A	7.85 ± 0.98	9.10 ± 0.80
	BKT Student Model	6.46 ± 0.78	6.65 ± 0.95	7.03 ± 1.00	6.82 ± 0.94	7.04 ± 0.96
	DKT Student Model	9.89 ± 1.45	8.69 ± 1.82	8.55 ± 2.08	8.31 ± 2.22	8.58 ± 2.13

Table 8.2: Robust evaluation matrix showing predictions for the five policies in our experiment according to the G-SCOPE student model as well as the BKT student model and a DKT student model. Notice that BKT-MP was not simulated on the new student model since they were not exactly compatible due to a nuance in the way they represent steps.

with 1,000 simulated students.

Using these off-policy policy performance estimates, the predicted Cohen's *d* effect size of AP-2 vs. Baseline 2 is 3.66 and the predicted effect size of AP-2 vs. Baseline 1 is 4.14, indicating that the new adaptive policies may yield a large improvement in robust student learning.

However, in our subsequent experiments there was no significant difference in the performance of students taught in the different policies as shown in Row 2 of Table 8.1.

We now consider the insight we could have obtained by using REM. We apply REM to our policies by evaluating them on three models: (1) the G-SCOPE model (which was used to derive AP-1 and AP-2), (2) the BKT student model (which was used to derive BKT-MP), and (3) a Deep Knowledge Tracing (DKT) model [Piech et al., 2015a]. All of these models were fit to the same data from our initial dataset from the semi-random instructional policy (i.e., data collected prior to running the experiment). The results are shown in Table 8.2.

Using the BKT student model, we see that all the policies appear to have much more similar expected performance than when using the G-SCOPE student model, though the new adaptive policies are still expected to be as good or better than the state-of-the-art BKT mastery policy in either situation, and an improvement over the non-adaptive policies. Therefore, were we only to simulate policies under the models used to derive the policies, we might still expect that the new adaptive policies would yield improved performance.

The key distinction comes up when we also simulate under another plausible student model, which was not used to derive a particular student policy. In contrast to the other student models, simulating using a Deep Knowledge Tracing student model actually predicts that Baseline 1 will yield the highest expected student learning performance, and be substantially higher than the predicted performance of the adaptive instructional policies.[3] Since three student models (BKT, G-SCOPE and DKT) are all seemingly reasonable choices of student models with similar predictive accuracies (RMSE between 0.41 and 0.44), our robust evaluation matrix suggests that we should not have been confident that new adaptive policies would yield a large effect size improvement over non-adaptive baselines or even necessarily be better than the non-adaptive

[3] This Deep Knowledge Tracing model was introduced by Piech et al. [2015a] after these experiments were conducted, so interestingly, we could not have done this analysis prior to running our experiment.

policies (thus consistent with the lack of difference in the true experimental results).

Therefore, in this case REM could have served as a diagnostic tool to identify that our new proposed adaptive policies might not yield the significant improvement we hoped for, by explicitly considering whether this improvement is robust across many plausible student models.

Using REM to Inform Policy Selection

We just presented a retrospective analysis of how REM *could have* informed our experiment had we used it before running the experiment. A natural next step was to see if REM could be used to actually discover a good instructional policy for our next experiment. Here we discuss how we used REM to discover a policy that we had confidence would outperform a baseline (namely, BP-1), and the results of the experiment that followed. Unfortunately, we again found no significant difference in terms of learning between the policy we chose using REM and the baseline policy. However, we discuss how this experiment combined with our REM analyses gave us new insights into the search for adaptive policies and how to do more robust analyses using REM.

So far we have been discussing how REM can help address the problem of wrong classes of student models. But notice that REM can also help address other related issues that may arise in educational contexts and certainly did arise in our first experiment. Recall that in the fractions tutor case study, the off-policy estimation was based on assuming students would do 40 problems each (i.e., we simulated trajectories of 40 problems). In reality, trajectories will be of varying length due to a number of factors: some students work faster than others, some students spend less time working or may be absent on certain days of our experiment, etc. However, even if we consider the variance in trajectory lengths that existed in our past data, the evaluation results would be similar. But one thing we did not consider is that the distribution of trajectory lengths varies for different instructional policies. For example, students who had the Baseline 1 policy, did around 48 problems on average, whereas for all the other policies, the average was 28 problems or less. This is, at least in part, because Baseline 1 only gives problems of a particular activity type (induction and refinement), which tended to be the activity type that took the least amount of time on average. This could explain why Baseline 1 did as well as the other policies in our experiment; these students simply had more problems, which could make up for the lack of diversity or adaptivity of problems. To tackle this problem, we can consider different generative models of how many problems students will do given a particular instructional policy (for example by taking into account how long problems took students in our past data); we can then use these various models as different student models (i.e., different rows in our matrix) and see if any policies robustly do well with respect to these differences. In what follows, each of our models assumed that the time per problem was sampled from how long students took in our prior data (and to increase robustness, we experimented with sampling times from different student populations that we had data for).

To see how important the time spent per problem might be, we tested a simple policy that sequenced problems in increasing order of average time students spent in our previous experiment (i.e. students would first get the problem that took the least amount of time on average). REM predicted that this policy would be better than the baseline induction and refinement policy under a variety of (but not all) student models. To make this policy adaptive, we augmented this policy with a simple rule to skip any problem where are skills taught in that problem were already believed to have been mastered (using a Bayesian Knowledge Tracing model with a mastery threshold of 0.9). We thought this might help avoid over-practice, especially because assigning problems in order of increasing time often meant giving similar problems multiple times in sequence. Indeed, this new adaptive policy was predicted by REM to be considerably better than the baseline according to many student models, including ones that predicted the non-adaptive version would be worse than the baseline. Models predicted the improvement of this new policy over the baseline would be between 0.31 and 2.23 points on the posttest, with most models predicting an improvement of at least 1 point on the posttest. Thus we chose to use this policy in our next experiment.

We ran an experiment with around 220 4th and 5th grade students to see if our new data-driven adaptive policy could outperform the baseline induction and refinement policy. Despite our REM predictions, when we ran our experiment, we found that students assigned the baseline policy had a mean posttest score of 8.12 (out of 16) and students assigned the new adaptive policy had a mean posttest score of 7.97, indicating the new policy was no better than the baseline. In terms of learning gains (posttest minus pretest score), the baseline had a mean score of 1.32, while the new adaptive policy had a mean scores of 1.55. While there was a positive difference, it was not significant. So one might ask, why did the new policy do worse than the induction and refinement baseline, when REM predicted otherwise?

There are two factors that we did not adequately account for in our REM analyses: (1) the student population in this experiment was quite different from the population in our past data that we used to fit the models, and (2) the order in which problems are presented was quite different than in our prior experiments. To account for the first issue, we had done REM analyses by fitting models to subpopulations of our prior data, but we had still predicted that the new adaptive policy would do better. We did more extensive analyses after the experiment, and we found that the predicted difference between the two policies was much smaller for students from a particular school district. Developing models and instructional policies that can generalize to new student populations is a big open question in the literature. While REM can help with this by seeing how different policies might interact with different populations of students we have collected data from, it cannot definitively tell us how the policy will effect with new students.

The second issue may have had an even greater effect on our results. All of the models that we used in REM assumed that the time per problem was sampled according to our prior data. Our new adaptive policy gave problems that took the least amount of time first, but it ignores the fact that students in our previous experiments had typically done those problems after having completed many other problems, which could be why they worked through those problems quickly. Indeed, in our experiment we found that problems given early on were taking students much longer than those same problems took for students in our previous experiment or in the baseline condition of the current experiment. Our experiment highlights the importance of not only modeling how students answer problems over time, but also how long they spend on problems, especially when we want to use time spent as a variable to determine how to adaptively assign problems to students. We believe future researchers can build on this insight in one of two ways:

1. developing more sophisticated ways of predicting how long students will spend on problems to use in offline analyses (such as REM analyses), or

2. developing policies that can be robust to how long students actually spend on problems by taking into account data collected from the student online.

We believe using REM with these insights can lead to the development of more robust instructional policies.

8.5.2 Case Study 2: Concept Learning

In Rafferty et al. [2015a], the authors consider three instructional policies for concept learning. The models are derived under three different partially observable Markov decision process (POMDP) student learning models of varying complexity inspired from the cognitive science literature: a memoryless model in which a learner maintains a single potential concept until evidence contradicts the correctness of this concept, a discrete model with memory which augments the memoryless model to prevent the learner from forgetting prior negative evidence about the potential concepts, and a continuous model which assigns probabilities to different potential concepts Rafferty et al. [2015a]. The model parameters were fit with data the authors collected from students given a random policy. The performance of a policy is measured in how long (time in seconds) it takes for students to learn a series of rules or a concept.

Like REM, the authors first simulate each policy on each of the three student models, but unlike REM, the authors only consider models that are used to derive some instructional policy (and no other student learning models). This is because the authors are interested in the interaction of student models with policies derived from student models and what that says about human learning, rather than using this simulation as an off-policy policy selection tool to help decide which instructional policies may offer a benefit over existing benchmarks. Indeed, Rafferty et al. test all policies with real students. We reinterpret their results in terms of insights REM would have offered about the relative expected performance among the policies.

In the first experiment, the authors find that in simulation, all three student models agree that the three policies induced by the POMDPs would enable student to learn the rules faster than a random policy (i.e., the memoryless, discrete with memory, and continuous policies do better than the random policy *in all three rows* of the robust evaluation matrix). We propose this should lead a practitioner to believe that these three policies will likely do better than a random policy when presented to actual students (if the student models are believed to be decent). Indeed, in their experiments, the authors found that all three POMDP policies induced a smaller average time to mastering the rules than the policy which selects activities randomly, two of which were statistically significantly faster.

In this situation REM consistently estimated that the adaptive policies would have higher performance than the random activity selection policies, under 3 different student models, and this result was confirmed experimentally. This shows a situation where REM consistently identified a predicted improvement, under a variety of student models.

We now consider another example from this work where REM could have helped predict that a policy would likely not work well in practice, but evaluating policies only under the models used to derive that policy would fail to identify this issue.

In their Experiment 3, Rafferty et al. compare various policies on three concept learning tasks both in simulation (under all three student models) and in an actual experiment. The following result is of most interest to us: when using the continuous POMDP model to simulate student learning, they find that a heuristic greedy policy derived from this model—the maximum information gain policy—does significantly better than both the random-action-selection policy and the two POMDP polices derived from other POMDP models. This was estimated to hold in all three concept learning tasks. However, in the actual experiment with students, the maximum information gain policy yields lower student performance than the random action selection policy and all the POMDP policies for all three concept learning tasks. This result could have been detected using REM, as both the memoryless model and the discrete model with memory estimated that the performance of the maximum information gain policy would be lower than the estimated performance of the random-action-selection instructional policy in at least one concept learning task. In this situation REM would have restricted the confidence with which one could expect the new policy to yield a big improvement in performance.

8.6 Discussion

In some cases, REM might result in one being overly-conservative by not deploying an instructional policy that is actually worthwhile. At the end of the day, it is up to each researcher to decide if they want to try a policy they think might result in improved student learning, even if not all models agree, or if they would rather find a policy they are confident would result in an improvement. One can attain such confidence (although not in any statistically precise sense) if one finds a policy that does very well under various student models as we saw an example of in Case Study 2. However, as we have emphasized several times, this confidence depends on being convinced that our choice of student models to use in the matrix was good. As we mentioned, we do not expect any of these student models to be correct, so what does it mean for a model to be “good”? A necessary condition is that such a model should be able to differentiate between different policies. For example, a model that predicts students are always in the same state (perhaps determined by their prior knowledge or pretest scores) and never learn would not be a good model to use in REM, because it would predict all instructional policies result in equal student outcomes. One way to avoid such “bad” models is to avoid models with bad predictive accuracy; even if high predictive accuracy is not a good indicator of a model’s ability to suggest good instructional policies, an especially low predictive accuracy should be a red flag. Effectively using REM can be thought of as a conversation between (potentially black box) machine learning algorithms and researchers who have to ultimately interpret what the results of the matrix say about the models and policies it is composed of and make the decision about when to use a certain policy.

As discussed in our first case study, effectively using REM involves considering not only different types of statistical models (such as Bayesian Knowledge Tracing, logistic regression models, and MDPs), but also models that can predict how long it takes students to solve problems, and models that are fit to specific sub-populations of students that we have data for. The issue of fitting models that accurately characterize how students learn across populations, and relatedly, finding policies that are robust to different student populations is an important open question in the learning sciences. To our knowledge, there have only been some investigations in this direction. For example, Clement et al. [2016] cast their work as evaluating algorithms that are optimized for some type of student (characterized by students with certain knowledge graphs) on simulated students that actually have different knowledge graphs. Their work can be viewed as using REM to explore the robustness of policies to different student populations.

Notice that REM could be used with a variety of objective functions. Here, we considered using REM to predict which instructional policy will lead to higher learning gains. However, REM could also be used to search for equitable policies. In this case, rather than having different student populations constitute different rows in REM, students from different populations could be considered in the same row, but the value in each cell will be some measure of the equity of the policy with respect to the different sub-populations. Indeed, this is what we implicitly did in the previous chapter when we compared various mastery learning policies (BKT, AFM-based mastery learning, and *N*-CCR) assuming students learn according to either a BKT or AFM model.[4] We found that AFM-based mastery learning was more equitable than BKT, but at the expense of giving extra practice. This suggests we may want to factor multiple objective functions in REM, including average learning gains, amount of extra practice, and equitability (e.g., difference in learning gains for different groups of students). Of course, this analysis just scratches the surface of using REM to find equitable knowledge tracing algorithms. Future work should expand the set of models and instructional policies considered to find more robust and equitable policies that positively impact student learning.

At this point we do not make any universal recommendations for how to use the robust matrix method to determine which instructional policy to use in the future. It is possible that one policy does not consistently do better than all other policies for every row of the matrix, but that it tends to do better, or that on average it does better. In this case, should we be confident in that policy? The answer must be determined on a case-by-case basis. The matrix might help reveal trends that can help the researcher determine whether a policy should be deployed or not. As mentioned earlier, it is not a black box algorithm that will tell the researcher what to do; it is a heuristic that can help inform the researcher to make better decisions. As we have shown, REM does not always work, but when it does not, it can lead us to consider what our models are missing, and

[4] However, notice that in the previous chapter, the models had hand-set parameters, while in this chapter, we only use models that were fit to student data. Fitting models to student data may result in models that are closer to reality (even though they are biased). But using hand-set parameters could also be informative, for example, when conducting sensitivity analyses to see how sensitive an instructional policy is to the parameters of a particular model.

can thus lead to advancements in student modeling and the search for content selection polices that are robust to model misspecification.

Chapter 9

Review of Reinforcement Learning for Instructional Sequencing*

> The development of a theory of instruction cannot progress if one holds the view that a complete theory of learning is a prerequisite. Rather, advances in learning theory will affect the development of a theory of instruction, and conversely the development of a theory of instruction will influence research on learning.
>
> RICHARD ATKINSON, 1972

We have seen that relying on individual models of learning can lead to misinformed decisions, while using high variance techniques like importance sampling can be too unreliable. Although these techniques have their limitations, one way to get a sense of which techniques might be most useful for instructional sequencing is to look to the past. In this chapter, I review previous attempts at empirically evaluating various uses of reinforcement learning for instructional sequencing. The primary conclusion I reach is that data-driven methods have been more successful when combined with psychologically-informed models of learning. I note that the idea of developing models informed by the learning sciences and psychology is complementary to using the robust evaluation matrix. While a single plausible model might lead to good instructional policies in some cases, REM can help provide more confidence that our policies are good. On the other hand, the more accurate our models are in capturing learning, the more useful REM will be. Figure 9.1 shows that various techniques to instructional sequencing that I discuss in this literature review. In what follows, I begin by presenting a brief historical review of attempts to use reinforcement learning for instructional sequencing followed by a review of the empirical literature.

*The work described here was done in collaboration with Vincent Aleven and Emma Brunskill.

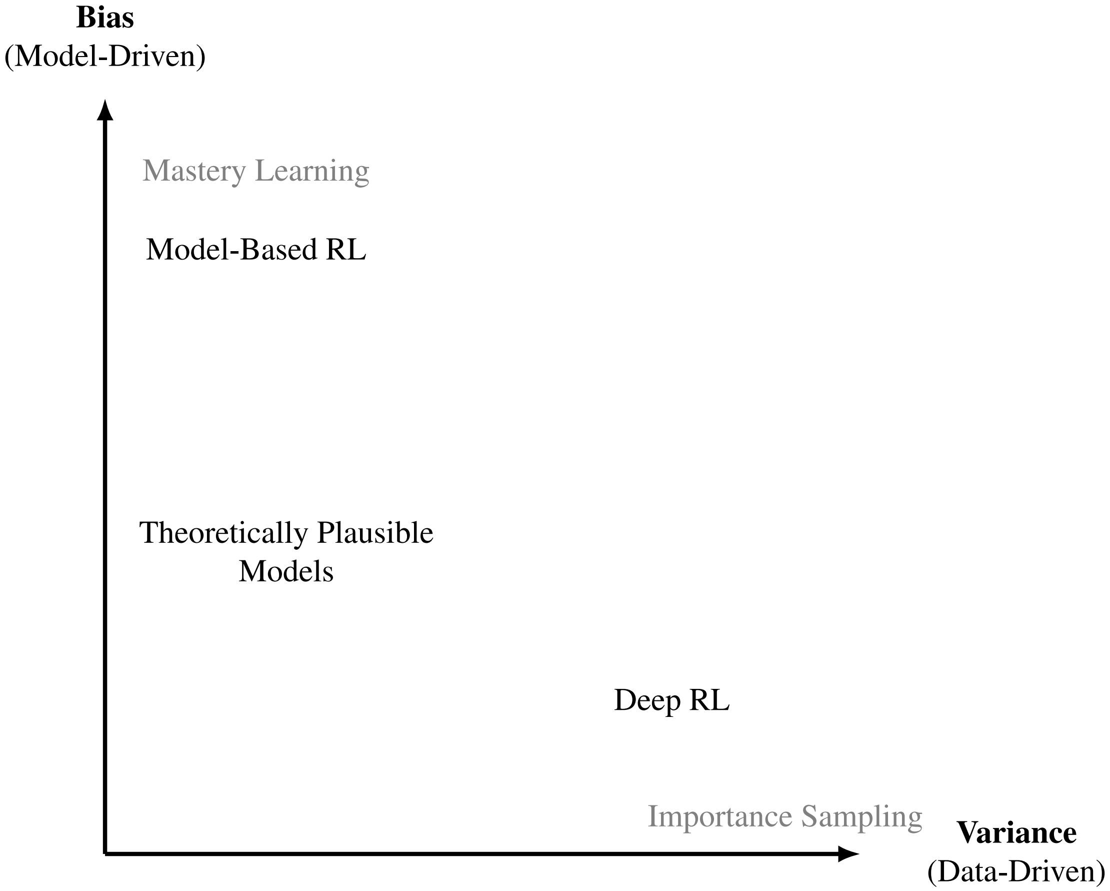


Figure 9.1: The techniques using RL applied to instructional sequencing discussed in this literature review (shown in black) positioned on the bias-variance tradeoff. One of the key findings in this chapter is that model-based RL techniques can be effective if they use models that are theoretically plausible (informed by psychology and the learning sciences).

9.1 A Historical Perspective

The use of reinforcement learning (broadly conceived) for instructional sequencing dates back to the 1960s. We believe at least four factors led to interest in automated instructional sequencing during the 60s and 70s. First, teaching machines (mechanical devices that deliver step-by-step instruction via exercises with feedback) were gaining a lot of interest in the late 50s and 60s, and researchers were interested in implementing adaptive instruction in teaching machines [Lumsdaine, 1959]. Second, with the development of computers, the field of computer-assisted instruction (CAI) was forming and there was interest in developing computerized teaching machines [Liu, 1960]. Third, pioneering work on mathematical optimization and dynamic programming [Bellman, 1957, Howard, 1960a], particularly the development of Markov decision processes, provided a mathematical literature for studying the optimization of instructional sequencing. Finally, the field of mathematical psychology was beginning to formulate mathematical models of learning [Atkinson and Calfee, 1963].

	More data-driven/data-generating ⟶		
	First Wave (1960s-1970s)	Second Wave (1990s-2010s)	Third Wave (2015-)
Instructional Technology	Teaching Machines/CAI	ITSs	MOOCs
Optimization Methods	MDP Planning	RL	Deep RL
Models of Learning	Mathematical Psychology	Machine Learning EDM/AIED	Deep Learning

Table 9.1: Trends in the three waves of interest in applying reinforcement learning to instructional sequencing. The "Instructional Tecnology" row shows technologies that were being developed or saw a lot of hype in the associated time period, even though older technologies were still used during the later time periods. The "Optimization Methods" row shows the form that RL research took in each time period; notice that the field of "reinforcement learning" was formally introduced in the late 1980s, but earlier work in MDP planning used with data-driven models in the 60s would still be considered RL. The "Models of Learning" row shows the research communities where new types of models of learning were emerging from during each time period.

As mentioned earlier, Ronald Howard, one of the pioneers of Markov decision processes, was interested in using decision processes to personalize instruction [Howard, 1960b]. In 1962, Howard's PhD student, Richard Smallwood, wrote his dissertation, *A Decision Structure for Teaching Machines* [Smallwood, 1962], which presented what is to our knowledge the first time an RL-induced instructional policy was tested on actual students. Even though the field of reinforcement learning had not yet developed, Smallwood was particularly interested in what we now call *online* reinforcement learning, where the system could improve over time. In fact, he provided preliminary evidence in his dissertation that the policy developed for his computerized teaching machine did in fact change with the accumulation of more data. Smallwood's PhD student Edward Sondik's dissertation, *The Optimal Control of Partially Observable Markov Decision Processes*, was seemingly the first text that formally studied planning in partially observable Markov decision processes (POMDPs). Sondik wrote in his dissertation, "The results obtained by Smallwood [on the special case of determining optimum teaching strategies] prompted this research into the general problem" [Sondik, 1971]. Thus, the analysis of POMDPs, an important area of research in optimal control, artificial intelligence, and reinforcement learning, was prompted by its application to instructional sequencing.

Around the same time, a group of mathematical psychologists at Stanford, including Richard Atkinson and Patrick Suppes, were developing models of learning from a psychological perspective and were interested in optimizing instruction according to these models, using the tools of dynamic programming developed by Howard and his colleagues. Atkinson and his students tested several instructional policies that optimized various models of learning [Atkinson, 1972b, Atkinson and Lorton, 1969, Chiang, 1974, Dear et al., 1967, Laubsch, 1969].

Curiously, there is almost no work on deriving optimal policies from the mid-70s to the early

2000s. While we cannot definitively say why, there seem to be a number of contributing factors. Researchers from the mathematical optimization community (including Howard and his students) stopped working on this problem after a few years and continued to work in their home disciplines. On the other hand, Atkinson left his career as a researcher in 1975 [Atkinson, 2014], and presumably the field of mathematical psychology lost interest in optimizing instructional policies over time. Research in automated instructional sequencing re-emerged at the turn of the twenty-first century for seemingly three reasons that completely parallel the trends that existed in the 60s. First, there was growing interest in intelligent tutoring systems, a natural testbed for adaptive instructional policies, paralleling the interest in teaching machines and computer-assisted instruction in the 60s. Second, the field of reinforcement learning formally formed in the late 1980s and early 1990s [Sutton and Barto, 1998], combining machine learning with the tools of Markov decision processes and dynamic programming built in the 60s. Finally, the field of Artificial Intelligence in Education (AIED) and, later, educational data mining (EDM) were interested in developing statistical models of learning, paralleling mathematical psychologists' interest in models of learning several decades earlier.

Even though there has been no void of research on instructional sequencing since the early 2000s, there seems to be a third wave of research appearing in this area in recent years. This is due to certain shifting trends in the research landscape that might be attracting a new set of researchers to the problem of data-driven instructional sequencing. First, there is a new "automated" medium of instruction, like the teaching machines, CAI, and ITSs of previous decades: MOOCs and other large-scale online education providers.[1] And with MOOCs comes the promise of big data. Second, the field of deep reinforcement learning has formed, leading to significantly more interest in the promise of reinforcement learning as a field. Indeed, there were around 35% more papers and books mentioning reinforcement learning in 2017 than in 2016 (as per the number of Google Scholar search hits). While initial advances in deep reinforcement learning have been focused largely on playing games such as Atari [Mnih et al., 2015] and Go [Silver et al., 2016, 2017], we have recently seen researchers applying deep reinforcement learning to the problem of instructional sequencing [Chaplot et al., 2016, Piech et al., 2015b, Reddy et al., 2017, Shen et al., 2018a, Upadhyay et al., 2018, Wang et al., 2017a]. Finally, in tandem with the use of deep reinforcement learning, there is also a growing movement within the AIED and EDM communities to use deep machine learning models to model human learning [Chaplot et al., 2016, Piech et al., 2015b]; this is a significantly different approach from the previous trends to use models that were more interpretable in the 1990s and models that were more driven by psychological principles in the 1960s.

Table 9.1 summarizes the trends that we believe have been responsible for the "three waves" of interest in applying reinforcement learning and decision processes to instructional sequencing. We find that there is a general trend that the methods of instructional sequencing have become more data-driven over time and the media for delivering instruction have become generally more

[1] Although many researchers are still testing RL-induced policies in ITSs and other platforms, there is reason to believe that MOOCs and other online instructional platforms such as Khan Academy and Duolingo have attracted many new researchers to this area. This can be witnessed by the emergence of Learning@Scale as a new conference that emerged as a result of MOOCs. Indeed, one of us (Doroudi) was drawn to AIED as a result of the development of MOOCs.

data-generating. Perhaps researchers are inclined to believe that more computational power, more data, and better reinforcement learning algorithms makes this a time where RL can have a demonstrable impact on instruction. However, we do not think these factors are sufficient for RL to leave its mark; we believe there are insights to gain about how RL can be impactful from the literature, which is where we will look to next. Based on the growth of interest in reinforcement learning in general and deep reinforcement learning in particular, we anticipate many more researchers will be interested in tackling instructional sequencing in the coming years. We hope this history and the review of empirical literature that follows will be informative to these researchers.

9.2 Review of Empirical Studies

To understand how successful RL has been in impacting instructional sequencing, I conduct a broad review of the empirical literature in this area. In particular I am interested in any studies that run a controlled experiment comparing one or more instructional policies, at least one of which is induced by an RL-based approach. The goal is to identify how often studies find a significant difference between RL-induced policies and baseline policies, and what factors might affect whether or not an RL-induced policy is successful in helping students learn beyond a baseline policy. Recent advances in reinforcement learning and educational technology, such as deep RL [Mnih et al., 2015] and big data, seem to be resulting in growing interest in applying RL to instructional sequencing. My hope is that this review will productively inform both researchers who are new to the field and researchers who are continuing to explore ways to impact instructional design with the tools of reinforcement learning.

9.2.1 Inclusion Criteria: Scope of the Review

One challenge of conducting this systematic review is determining what counts as an "RL-induced policy." First of all, not all studies (especially ones from the 60s and 70s) use the term reinforcement learning, but they are clearly doing some form of RL or at least applying Markov decision processes to the task of instructional sequencing. Second, some studies are not clearly conducting some form of RL, but still have the "flavor" of using RL in that they find instructional policies in a data-driven way or they use related techniques such as multi-armed bandits or Bayesian optimization. On the other hand, some studies that *do* use the language of RL use many heuristics or approximations in trying to find an instructional policy (such as myopic planning). Our goal was to include all studies that had the "flavor" of using RL-induced instructional policies, even when the language of RL or related optimization techniques are not used.

The challenge with this is that there are two components to reinforcement learning: (1) optimization (e.g., MDP planning in the model-based setting) and (2) learning from data (e.g., learning the MDP in the model-based setting). Many studies emphasized one of these but not the other. We determined that for a study to be considered as using RL for instructional sequencing, it should use some form of optimization and data to find instructional policies, although we wanted to be fairly inclusive in how optimization procedures and data were used. Our goal was to be inclusive while not allowing any studies that simply used heuristics or expert-determined rules to derive adaptive policies. More formally, we consider any studies where:

- The study acknowledges (at least implicitly) that there is a model governing the learning process, and that giving different instructional actions to a student might probabilistically change the state of a student according to the model.
- There is an instructional policy that maps past observations from a student (e.g., responses to questions) to instructional actions.
- Data collected from students (e.g., correct or incorrect responses to previous questions), either in the past (offline) or over the course of the study (online), are used to learn either:
    - the model, or
    - the instructional policy
- If the model is learned, the instructional policy is designed to approximately optimize that model according to some reward function, which may be implicitly specified.

Notice that this means we consider any studies that might learn a model from prior data and then use a heuristic approximation (such as myopic planning rather than long-horizon planning) to find the instructional policy. While it might make sense to also include any studies that applied planning to a pre-specified MDP or POMDP (without learning any parameters), we chose to only include studies that involve at least some learning from data, as we believe that is a critical component of reinforcement *learning*.

Searching for all papers that match our inclusion criteria is challenging as not all papers use the same language to discuss data-driven instructional sequencing (e.g., not all papers refer to reinforcement learning, Markov decision processes, etc.). Therefore, to conduct our search, we began with an initial set of papers that we knew matched our inclusion criteria, and iteratively added more papers by performing one-step forward and backward citation tracing. That is, for every paper that we included in our review, we looked through the papers that it cited as well all papers that cited it (as identified by Google Scholar) to see if any of those papers also match our inclusion criteria. This means if we have missed any relevant studies, they are completely disconnected (in terms of citations) from the studies that we have identified. We found relevant papers coming from a diversity of different research communities including mathematical psychology, cognitive science, optimal control, AIED, educational data mining, machine learning, and human-robot interaction. While it is possible that there are papers from communities outside of these, we assume that most papers would have at least cited one of the papers we identified.

9.2.2 Results

We found 34 papers containing 41 studies that matched our criteria, including a previously unpublished study that we ran on our fractions tutoring system, which is described in Section 8.5.1. Before discussing these studies in depth, it is worth briefly mentioning the kinds of papers that did not match our inclusion criteria, but are still related to studying RL for instructional sequencing. Among these papers, we found 19 papers that learned policies on offline data but did not evaluate the performance of these policies on actual students.[2] At least an additional 23 papers learned (and compared) policies using only simulated data (i.e., no data from real learners were used).[3] Moreover, we found at least eight papers that simply proposed using RL for instructional sequencing or proposed an algorithm for doing so in a particular setting without actually providing concrete results.[4] We also found at least fourteen papers that did use instructional policies on real students, but did not match our inclusion criteria for various reasons, including not being experimental, varying more than just the instructional policy across conditions, or using hand-set model parameters.[5] For example, Corbett and Anderson [1995] compare using a model (BKT) to determine how many remediation exercises should be given for each KC, but they compare this to providing no remediation, not another way of sequencing remediation exercises. Finally, many papers have formally studied optimal instructional policies under various models of learning, especially during the first wave of optimizing instructional sequencing [e.g. Karush and Dear, 1967, Smallwood, 1968, 1971].

The various papers that study RL-induced instructional policies outside of empirical evaluations are interesting in their own right; however, they are not the focus of this review, because they do not give us concrete evidence about how well RL-induced instructional policies would perform in practice. If nothing else, the sheer number of papers that study RL-induced policies in one form or another shows that there is broad interest in applying RL to instructional sequencing, especially as these papers come from a variety of different research communities. However, less than half of the papers that explore this area actually test various methods of instructional sequencing on actual learners. One reason for this is surely because it is much harder to actually

[2] These papers include: [Chi et al., 2008], [Theocharous et al., 2010], [Mitchell et al., 2013b], [Mitchell et al., 2013a], [Mota et al., 2015], [Piech et al., 2015b], [Rollinson and Brunskill, 2015], [Lan and Baraniuk, 2016], [Hoiles and Schaar, 2016], [Antonova et al., 2016], [Käser et al., 2016], [Chaplot et al., 2016], [Lin and Chi, 2016], [Shen and Chi, 2016a], [Wang et al., 2016], [Wang et al., 2017a], [Sawyer et al., 2017], [Tabibian et al., 2017], and [Fenza et al., 2017].

[3] These papers include: [Chant and Atkinson, 1973], [Iglesias et al., 2003], [Martin and Arroyo, 2004], [Sarma and Ravindran, 2007], [Iglesias et al., 2009], [Theocharous et al., 2009], [Folsom-Kovarik et al., 2010], [Kujala et al., 2010], [Champaign and Cohen, 2010], [Malpani et al., 2011], [Pietquin et al., 2011], [Daubigney et al., 2013], [Dorça et al., 2013], [Schatten et al., 2014], [Andersen et al., 2016], [Clement et al., 2016], [Reddy et al., 2017], [Goel et al., 2017], [Wang et al., 2017b], [Zaidi et al., 2017], and [Mu et al., 2018], [Upadhyay et al., 2018], [Lakhani, 2018].

[4] These papers include: [Beck, 1997], [Bennane et al., 2002], [Legaspi and Sison, 2002], [Almond, 2007], [Brunskill and Russell, 2011], [Ramachandran and Scassellati, 2014], [Mejía-Lavalle et al., 2016], and [Spaulding and Breazeal, 2017].

[5] These papers include: [Smallwood, 1962], [Corbett and Anderson, 1995], [Joseph et al., 2004], [Pavlik et al., 2008], [Iglesias et al., 2006], [Van Rijn et al., 2009], [Nijboer, 2011], [Folsom-Kovarik, 2012], [Lomas et al., 2012], [Wang, 2014], [Leyzberg et al., 2014], [Settles and Meeder, 2016], [Sense, 2017], and [Hunziker et al., 2018].

test methods of instructional sequencing on actual students than to do experiments in simulation; this is especially true when running experiments in authentic classroom settings. We now turn our attention to the set of studies that *do* evaluate instructional policies with students.

For the studies that met our inclusion criteria, the first row of Table 9.2 shows how the studies are divided in terms of varying "levels of significance." Eighteen of the 36 studies found that at least one RL-induced policy was statistically significantly better than all baseline policies for some outcome variable, which is typically performance on a posttest after engaging with the instructional policy. Four studies found no significant difference overall, but a significant aptitude-treatment interaction (ATI) favoring low-performing students (i.e., finding that the RL-induced policy performed significantly better than the baselines for lower performing students but no significant difference was detected for high performing students). Three studies found mixed results, namely that an RL-induced instructional policy outperformed at least one baseline policy but not all the baselines. Ten studies found no significant difference between adaptive policies and baseline policies. Finally, only one study technically found that a baseline policy outperformed an RL-induced policy.[6]

Thus, over half of the studies found that adaptive policies outperform all baselines that were tested. Moreover, the studies that found a significant difference, as well as those that demonstrated an aptitude-treatment interaction, often found a Cohen's d effect size of at least 0.8, which is regarded as a large effect [Cohen, 1988]. While this is seemingly a positive finding in favor of using RL-induced policies, it does not tell us *why* some studies were successful in showing that RL-induced policies can help students learn beyond a baseline policy, and why others were less successful. In order to obtain a better understanding of when and where RL has been successful when used for instructional sequencing, we qualitatively cluster the studies into five different groups based on how they have applied RL. The clusters generally vary in terms of the types of instructional actions considered and how they relate to each other. For example, in paired-association tasks, each action specifies the content presented to the student, but each piece of content is assumed to be independent of the rest. In the sequencing interdependent content cluster, each action also specifies the content, but the various pieces of content are assumed to be interdependent. On the other hand, in the sequencing activity types cluster, each possible action specifies the type of instructional activity, not the content.

For each cluster, we provide a table that gives a summary of all of the studies in that cluster and we describe some of the key commonalities and differences among the studies. In doing so, we will (a) demonstrate the variety of ways in which RL can be used to sequence instructional activities, and (b) set the stage for obtaining a better understanding of the conditions under which RL has been successful in sequencing instruction for students, which we discuss in the next section. Table 9.2 shows each of the five clusters of studies that we describe below, along with the number of studies in each cluster in each "significance level" that we identified above (e.g., whether the study showed a significant effect in favor of RL-induced policies, an aptitude-treatment interac-

[6] As we discuss later, the "baseline" in this case was actually an adaptive policy that the authors developed—Adaptive Response-Time-based sequencing (ARTS) [Mettler et al., 2011]. However, since the ARTS model is not fit to data, it does not satisfy our criteria for an RL-induced adaptive policy, whereas the policy they compared to was based on the model from Atkinson [1972b] and was fit to data.

	Sig	ATI	Mixed	Not Sig	Sig Worse
All Studies	20	4	5	11	1
Paired-Association Tasks	11	0	0	2	1
Concept Learning Tasks	3	0	3	1	0
Sequencing Interdependent Content	0	0	2	6	0
Sequencing Activity Types	4	4	0	2	0
Not Optimizing Learning	2	0	0	0	0

Table 9.2: Comparison of clusters of studies based on the "significance level" of the studies in each cluster: **Sig** indicates that at least one RL-induced policy significantly outperformed all baseline policies, **ATI** indicates an aptitude-treatment interaction, **Mixed** indicates the RL-induced policy significantly outperformed some but not all baselines, **Not sig** indicates that there were no significant differences between policies, **Sig worse** indicates that the RL-induced policy was significantly worse than the baseline policy (which for the only such case was an adaptive policy).

tion etc.). The table clearly shows that the different types of studies had very varying levels of success. Therefore, a qualitative understanding of each cluster will help us understand when and where RL can be most useful for instructional sequencing. Appendix B provides more details about the particulars of all the studies, including the types of models and instructional policies used in these studies.

Paired-Association Tasks

The studies in this cluster are listed in Table 9.3. Notice that all of the studies that were run in the first wave of instructional sequencing (1960s-70s) belong to this cluster. A paired-association task is one where the student must learn a set of pairs of associations. The most common type of paired-association task (and one which is of educational value) is learning a set of vocabulary words in a foreign language. In such tasks, a stimulus (e.g., a foreign word) is presented to the student, and the student must attempt to provide the translation of the word. The student will then see the correct translation. Such tasks may also be referred to as flashcard learning tasks, because the student is essentially reviewing a set of words or concepts using "flashcards." A key assumption in any paired-association task is that the stimuli are independent of one another. For example, if one learns how to say "chair" in Spanish, it is assumed that it does not help or hinder one's ability to learn how to say "table" in Spanish. Because of this assumption, we may also think of this cluster as "sequencing independent content," which clearly contrasts it with some of the later clusters.

The key goal in sequencing instruction for paired-association tasks is to balance between (1) teaching stimuli that the student may have not learned yet, and (2) reviewing stimuli that the student may have forgotten or be on the verge of forgetting. The psychology literature has shown that sequencing instruction in such tasks is important due to the presence of the spacing effect

Paper(s)	Domain	Population	Setting	# of Actions	# of Subjects	Effect
Laubsch [1969]	Swahili-English	Uni	Lab	$\binom{140}{35}$	24	Sig
Atkinson and Lorton [1969]	English Spelling	Gr 4-6	Lab	24	42	Sig
Atkinson [1972b]	German-English	Uni	Lab	12	30	Sig
Chiang [1974] Exp 1	Chinese-English	Uni	Lab	$\binom{84}{21}$	12	Sig
Chiang [1974] Exp 2	Chinese-English	Uni	Lab	$\binom{84}{21}$	12	Sig
Katsikopoulos et al. [2001] Exp 1	String-Num Mapping	Adults	Lab	12	16	Sig
Pavlik and Anderson [2008]	Japanese Words	Adults	Lab	180	20	Sig
Lindsey et al. [2014b]	English-Spanish	Gr 8	Class	221	57	Sig
Lindsey [2014]	English-Spanish	Gr 8	Class	221	56	Sig
Papoušek et al. [2016]	Geography	Online	Online	?	≈5000	Sig
Leyzberg et al. [2018]	Spanish-English	Gr 1	Lab	4	9	Sig
Dear et al. [1967]	Num-Num Mapping	Uni	Lab	16	40	Not sig
Katsikopoulos et al. [2001] Exp 2	String-Num Mapping	Adults	Lab	24	12	Not sig
Mettler et al. [2011]	African Countries	Uni	Class	24	50	Sig worse

Table 9.3: Summary of all empirical studies in the paired-association tasks cluster. The number of subjects column reports the number of subjects in the condition with the least subjects. A '?' indicates something that we could not deduce from the paper.

[Ebbinghaus, 1885], whereby repetitions of an item or flashcard should be spaced apart in time. Thus, a key component of many of the instructional policies developed for paired-association tasks is using a model of forgetting to predict the optimal amount of spacing for each item. Early models used to sequence instruction such as the One-Element Model (OEM) ignored forgetting and only sequenced items based on predictions of whether students had learned the items or not [Bower, 1961, Dear et al., 1967], but Atkinson [1972b] later developed the Markov model we described in Section 5.2.1, which accounted for forgetting, and he showed that it could be used to successfully sequence words in a German to English word translation task. More recently, researchers have developed more sophisticated psychological models that account for forgetting such as the Adaptive Control of Thought—Rational (ACT-R) model [Anderson, 1993, Pavlik and Anderson, 2008], the Adaptive Response Time based Sequencing (ARTS) model [Mettler et al., 2011], and the DASH model [Lindsey et al., 2014b]. See Appendix B.1 for a brief description of these models. In some of the studies, policies using these more sophisticated models were shown to outperform RL-induced policies that used Atkinson's original memory model [Mettler et al., 2011, Pavlik and Anderson, 2008].

Thus, aside from the type of task itself, a key feature of the studies in this cluster is their use of psychological models. As shown, in Table 9.2, only 2 studies showed no significant difference between RL-induced policies and baseline policies. In both of these studies [Dear et al., 1967, Katsikopoulos et al., 2001], the model that was used was the OEM model—a simple model that does not account for forgetting and hence cannot space instruction of paired associate over time. Similarly, Laubsch [1969] compared two RL-induced policies to a random baseline policy, and found that the policy using the OEM model did not do significantly better than the baseline while the policy based on the more sophisticated Random-Trial Increment (RTI) model did better. Fi-

Paper(s)	Domain	Population	Setting	# of Actions	# of Subjects	Effect
Rafferty et al. [2011]	Alphabet Arithmetic	Online	Online	45	20	Sig
Rafferty et al. [2016a] Exp 1	Alphabet Arithmetic	Uni	Lab	45	20	Sig
Sen et al. [2018]	Chemical Molecules	AMT	Online	71	100	Sig
Lindsey et al. [2013]	Concept Learning	AMT	Online	256	50	Mixed
Rafferty et al. [2016a] Exp 2	Number Game	Uni	Lab	300	20	Mixed
Whitehill and Movellan [2017]	Picture-Word Mapping	AMT	Online	16997	26	Mixed
Geana [2015]	Concept Learning	Uni	Lab	216	25	Not sig

Table 9.4: Summary of all empirical studies in the concept learning tasks cluster

nally, the only study that showed a baseline policy significantly outperformed an RL-induced policy, was the comparison of a policy based on the ARTS model with a policy based on the model used by Atkinson [1972b]. The ARTS model was actually a more sophisticated psychological model than Atkinson's, but the parameters of the model were not learned from data, and therefore, we considered the policy based on ARTS to technically be a non-RL-induced "baseline" policy.

Concept Learning Tasks

Concept learning is another type of task where several researchers have applied RL-based instructional sequencing. The studies in this cluster are shown in Table 9.4. Concept learning tasks are typically artificially-designed tasks that can be used to study various aspects of human cognition, and as such are commonly studied in the cognitive science literature. In a concept learning task, a student is presented with examples that either belong or do not belong to an initially unknown concept, and the goal is to learn what constitutes that concept (i.e., how to distinguish between positive examples that fit the concept and negative examples that do not). This is similar to a classification task in machine learning. For example, Rafferty et al. [2016a] use a POMDP to sequence instructional activities for two types of concept tasks, one of which is called the Number Game [Tenenbaum, 2000], where students see numbers that either belong or do not belong to some category of numbers such as multiples of seven or numbers between 64 and 83. While such tasks are artificial and of little direct educational value, the authors' goal was to show that models of memory and learning concepts from cognitive psychology could be extended to a POMDP framework to teach people concepts quickly, which they succeeded in doing. Whitehill and Movellan [2017] extend this idea to teaching a concept learning task that is more authentic: learning foreign language vocabulary words via images. Whitehill and Movellan [2017] call this a "Rosetta Stone" language learning task, as it is inspired by the popular language learning software, Rosetta Stone. Notice that this task differs from teaching vocabulary as a paired-association task, because there are multiple images that might convey the meaning of a foreign word (i.e., the concept), and the the goal is to find a policy that can determine at any given time both what foreign vocabulary word to teach and what image to present to convey the meaning of that word. Sen et al. [2018] also use instructional policies in an educationally-relevant concept learning task, namely perceptual fluency in identifying if two chemical molecules shown in different representations are the same. In this task, the student must pick up features of the representations that help identify which chemical molecule is being shown.

Unlike paired-association tasks, the various pieces of content that can be presented in a concept learning task are mutually interdependent, but in a very particular way. That is, seeing different (positive or negative) examples for a concept help refine one's idea of a concept over time.[7] For example, in the Number Game, knowing that 3, 7, and 11 are in a concept might lead one to think the concept is likely odd numbers, while also knowing that 9 is not a member of the concept might lead one to believing the concept is likely prime numbers. The exact sequence of examples presented can have a big influence on a student's guess as to what the correct concept might be. Therefore, determining the exact sequence of examples to present is critical for how to most quickly teach a given concept. Moreover, in these tasks, it is often beneficial to use information-gathering activities (e.g., giving a quiz to test what concept the student finds most likely), to determine what examples the student needs to refine their understanding.

As with the paired-association task studies, one common feature among the studies in this cluster is that they have typically used psychologically inspired models of learning coming from the concept learning literature and computational cognitive science literature. To illustrate an example of how a psychological model of learning could be used to sequence instruction for students, we consider the POMDP framework used by Rafferty et al. [2016a], where the authors used three psychological models of varying degrees of complexity. Rafferty et al. [2016a] consider giving three types of actions for teaching a concept: examples, quizzes, and questions with feedback. An example action presents an example that can help one infer the concept (e.g., a number that belongs to the concept or one that does not). A quiz action will ask the student a question to assess whether or not they understand a concept. Finally, a question with feedback action combines the two. Here we consider the simplest POMDP the authors considered, which was based on a model of concept learning from Restle [1962]. This POMDP assumes that the student initially thinks a particular concept is the correct one given some prior over concepts (e.g., the student might be much more likely to think the concept is odd numbers or multiples of 4 rather than an arbitrary collection of numbers). Every time the student is given an example, the student will either maintain their current concept if it is consistent with the example, or switch to thinking another concept is the correct one, randomly choosing a concept that is consistent with the most recent example according to their prior distribution over concepts. If the student is given a quiz, the student will generally answer according to the concept they believe in, with some small probability of giving an answer inconsistent their concept. Each action incurs a cost (or negative reward) based on the amount of time that action tends to take until the student has learned the correct concept, where there is no cost. Thus, the optimal policy will try to maximize reward (or minimize cost) by teaching the correct concept in the least amount of time. Again, since POMDP planning is in general intractable, Rafferty et al. [2016a] performed two-step lookahead planning

[7] Despite this difference between concept learning tasks and paired association tasks, one of the concept learning tasks used by Rafferty et al. [2011, 2016a], Alphabet Arithmetic, is actually quite similar to paired-association tasks, in that the goal is for students to learn a mapping of letters (A-F) to numbers (1-6), which is the concept to be learned. What distinguishes the task from paired-association tasks is that the examples shown to the learner aren't mappings but rather arithmetic equations (e.g., A + B = 5). Therefore, each example gives some information about two letter-number mappings.

Paper(s)	Domain	Population	Setting	# of Actions	# of Subjects	Effect
Green et al. [2011] Exp 1	Finite Field Arithmetic	Uni	Lab	?	26	Mixed
Green et al. [2011] Exp 2	Artificial Language	Uni	Lab	?	5	Mixed
Clement et al. [2015]	Arithmetic (with Coins)	7-8yo	Class	≥ 28	133	Not sig
David et al. [2016]	Basic Math	K-12	Class	?	35	Not sig
Schatten [2017]	Basic Math	K-12	Class	?	49	Not sig
Doroudi et al. [2017a]	Fractions	Gr 4-5	Class	155	69	Not sig
Section 8.5.1	Fractions	Gr 4-5	Class	155	100	Not sig
Segal et al. [2018]	Math	Gr 7	Class	?	9	Not sig

Table 9.5: Summary of all empirical studies in the sequencing interdependent content cluster

(where at each step, the agent acts as though it is only going to take two more actions).

Sequencing Interdependent Content

This cluster focuses on sequencing content, under the assumption that different areas of content are interdependent. The studies in this cluster are shown in Table 9.5. The sequencing task here is closest to traditional "curriculum sequencing," or ordering various content areas for a given topic. However, unlike traditional curriculum sequencing, the ordering of content can be personalized and adaptive, for example based on how well students have mastered various pieces of content. While concept learning tasks also have interdependent content, the goal in concept learning tasks is to teach a single underlying concept and the various pieces of content are simply examples that do or do not belong to that concept. In this cluster, the goal is to teach a broader scope of content under the assumption that how the content is sequenced affects students ability to learn future content. An instructional policy in this context must implicitly answer questions like the following: When teaching students how to make a fraction from the number line, when should we move on to the next topic and what should that topic be? Should the next topic depend on how well the student answered questions about the number line? If the student is struggling with the next topic, should we go back and teach some prerequisites that the student might have missed? When should we review a content area that we have given the student previously?

For these studies, typically either a network specifying the relationship between different content areas or KCs (such as a prerequisite graph) must be prespecified or such a network must be automatically inferred from data. Section 8.5.1 describes one of our studies performed in a fractions tutoring system where the relationships between different KCs was automatically inferred from data. As we see from Table 9.2, the studies in this cluster have been the least successful, with all of them resulting in either a mixed result or no significant difference between policies. We analyze why this might be in the next section.

Paper(s)	Domain	Population	Setting	# of Actions	# of Subjects	Effect
Chi et al. [2010a,b]	Physics	Uni	Lab	4	29	Sig
Lin et al. [2015] Exp 1	Linear Algebra	Uni	Lab	3	13	Sig
Lin et al. [2015] Exp 2[a]	Linear Algebra	Uni	Lab	3	12	Sig
Zhou et al. [2017]	Probability	Uni	Class	2	77	Sig
Shen and Chi [2016b]	Logic	Uni	Class	2	33	ATI
Shen et al. [2018a] Exp 1	Logic	Uni	Class	2	37	ATI
Shen et al. [2018a] Exp 2	Logic	Uni	Class	2	34	ATI
Shen et al. [2018b]	Logic	Uni	Class	2	39	ATI
Chi et al. [2009]	Physics	Uni	Lab	2	37	Not sig
Rowe [2013], Rowe and Lester [2015]	Microbiology	Gr 8	Class	2-6	28	Not sig

[a] The distinguishing factor between Exp 2 and Exp 1 by Lin et al. [2015], is that in Exp 2, the subjects did not have prior knowledge in the domain while in Exp 1, they did have prior knowledge.

Table 9.6: Summary of all empirical studies in the sequencing activity types cluster

Sequencing Activity Types

While the previous three clusters of studies were based on the way various pieces of content did or did not depend on each other—this cluster is about how to sequence the types of activities students engage with rather than the content itself. The studies in the sequencing activity types cluster are shown in Table 9.6. These studies used RL to determine what activity type to give at any given time for a fixed piece of content, based on the content being taught and the work that the student has done so far. For example, Shen and Chi [2016b], Zhou et al. [2017], Shen et al. [2018a], and Shen et al. [2018b] all consider how to sequence worked examples and problem solving tasks. Similarly, Chi et al. [2009, 2010a,b] consider, for each step, whether the student should be told the solution or whether the student should be asked to provide the solution, and, in either case, whether the student should be asked to justify the solution. Notice that these studies consider using RL for step-loop adaptivity as opposed to task-loop adaptivity, which all of the other studies reported in this review consider.

In the concept learning tasks described above, the RL agent must also decide how to sequence different activity types (examples, quizzes, and questions with feedback), but in those cases, the content of each activity also had to be selected (e.g., if the agent wants to present an example, it must also decide *which* example of the concept to present).

For the studies that use RL to sequence worked examples and problem solving tasks, we note that the existence of an expertise-reversal effect [Kalyuga et al., 2003], where novices benefit more from reviewing worked examples while experts benefit more from problem solving tasks. This suggests an ordering where worked examples are given prior to problem solving tasks (for learners who are initially novice). Renkl et al. [2000] have further shown that fading steps of worked examples over time, such that students have to fill-in incomplete steps of worked examples until they solve problems on their own, is more beneficial than simply pairing worked examples with problem solving tasks. Thus, in this setting, we know that the sequence of instructional activities can make a difference, which could explain why many of the studies that sequence worked

Paper(s)	Domain	Population	Setting	# of Actions	# of Subjects	Effect
Beck et al. [2000]	Arithmetic	Gr 6	Class	≥ 100	39	Sig
Mandel et al. [2014b]	Fractions	Kids	Game	7	500	Sig

Table 9.7: Summary of all empirical studies in the not maximizing learning cluster

examples and problem solving tasks either found a significant improvement above a baseline or an aptitude-treatment interaction. The studies in this cluster always compared to a policy that randomly sequences worked examples and problem solving tasks. Thus, it is not known if the RL-induced adaptive policies explored in this cluster would do better than a heuristic suggested by the expertise-reversal effect. Future work in this area is needed to determine whether RL is useful in inducing adaptive policies for sequencing activity types beyond heuristic techniques, or if RL can simply help find one of many decent policies that can outperform randomly sequencing activity types.

Not Maximizing Learning

There are two studies that do not fit into any of the previous four clusters, because they do not optimize for how much or how fast students learn. Beck et al. [2000] sequence instructional activities in an intelligent tutoring system with the goal of minimizing the time spent per problem, which their resulting policy achieved. While minimizing the time per problem could result in teaching students faster, it could also lead to the policy choosing instructional activities that are less time consuming (but not necessarily beneficial for student learning). Mandel et al. [2014b] try to maximize the number of levels completed in an educational game, and their RL policy does significantly increase the number of levels completed over both a random policy and an expert-designed baseline policy. While interesting, these two papers do not shed light on whether RL can be used to significantly improve student learning over strong baseline policies.

9.3 Discussion: Where's the Reward?

We now turn to analyzing what the results of the systematic review tell us about how impactful RL has been in the domain of instructional sequencing, and when and where it might be most impactful. Our results can be used to present both a pessimistic and an optimistic narrative about the success of RL in instructional sequencing. We will describe each of these narratives in turn. These narratives only capture some of the story of the impact RL has had in instructional sequencing. Thus, we will conclude this section by listing some other considerations that may have influenced when and where RL has been successful in sequencing instructional activities.

9.3.1 The Pessimistic Narrative: RL is Useful for Constrained Tasks

As for the pessimistic narrative, our results suggest that RL has seemingly been more successful in more constrained and limited settings. For example, the cluster where RL has been most successful is paired-association tasks. While paired-association tasks are relevant to teaching lists of words or facts, they do not extend to more complex, educationally relevant instruction. RL has also been relatively successful in concept learning tasks, but there has not yet been a demonstration of these tasks as designed for typical classroom learning activities. Finally, RL has been relatively successful in sequencing activity types, which *is* an interesting problem that educators and sophisticated learning technologies must able to tackle. But perhaps the success of RL in this area is due to the fact that it must only choose between two to four actions at any given time step.

However, when it comes to sequencing interdependent content, where the agent must choose one of many possible actions, there is not yet evidence that RL can induce instructional policies that are significantly better than reasonable baselines. This could be in part due to the fact that under the assumption that content is interrelated, the state of the MDP or POMDP could depend on the entire sequence of content given to the student so far. Thus the "true" state space could be exponential in terms of the horizon (i.e., number of decisions that need to be made for a student), even if we choose to model the state space in smaller terms. This means we might need an inordinate amount of data to learn a decent policy. Moreover, in order to posit whether problem A or problem B should be presented first, we need to have some data where A was presented before B and data where B was presented before A. In general, this means we need data from randomly presenting different orders of content to students. But if the content is indeed interrelated, presenting content in a random order may be difficult to justify, and may lead to inadequate learning.

9.3.2 The Optimistic Narrative: Learning Theory Helps

Another way of looking at the results paints a more optimistic picture. As mentioned earlier, for both paired-association tasks and concept learning tasks, the models that were used were informed by the psychology literature. Moreover, in the case of paired-association tasks, we noted that as psychological models got more sophisticated over time, the result of using them to induce instructional policies also got more successful, to the point that policies from more sophisticated psychological models sometimes outperformed policies from less sophisticated models (see Section 9.2.2 for more details). We also noted that an instructional policy derived from the ARTS model (a psychological model that was not fit to data) outperformed an instructional policy derived from the data-driven model developed by Atkinson [1972b]. Thus, in some cases a good psychological theory might be more useful for finding good instructional policies than a data-driven model that is less psychologically plausible. On the other hand, for sequencing activity types and interdependent content, the models used were solely data-driven.

Moreover, we found that for paired-association tasks and sequencing activity types, there are well-known results from psychology and the learning sciences that shows sequencing matters:

the spacing effect [Ebbinghaus, 1885] and the expertise-reversal effect [Kalyuga et al., 2003] respectively. On the other hand, for sequencing interdependent content, we do not have domain-general principles from the learning sciences that tell us whether and how sequencing matters. While it seems natural that sequencing different content areas makes a difference, especially when some content have strong prerequisites, this level of difference might not be detectable in short-term interventions. For example, when students are exposed to a new content area for the first time, they might need to study that content area for a long time before moving on. Therefore, a mastery learning policy with a good prerequisite graph along with direct instruction that clearly introduces each new concept might be more effective than adapting the sequence of concepts at a fine-grained level without providing thorough instruction whenever a new concept is introduced. On the other hand, if we are giving students remedial exercises for content they have already seen before, it might not matter much how we order different content areas. In this case, a policy that reasons about how to space different KCs to avoid forgetting (as was used for paired-association tasks) might be more useful than reasoning about the interrelationship of different content areas.

Thus, psychology and the learning sciences can give us insights for both how to make RL more likely to succeed in finding good instructional policies as well as when to hypothesize the precise sequencing of instructional activities might matter. These two competing narratives we have just presented are not wholly unrelated. The settings which are better studied by psychologists and where we have better theories and principles to rely upon are often more constrained, because such settings are easier for psychologists to tackle. But this does not mean RL should only be used in simple, unrealistic settings. Rather, it suggests that we should leverage existing theories and principles when using RL, rather than simply taking a data-driven approach. We explore this idea further in Section 9.4.

9.3.3 Additional Considerations

Other than the factors considered in the two narratives above, there are several other factors that could impact the success of RL in instructional sequencing.

Prior Knowledge

RL may have more room for impact in domains when students are learning material for the first time, because students have more room to learn. Almost all of the paired-association tasks are in domains when students have never learned the material before, such as foreign language learning. In many of these studies, researchers specifically took students who did not have expertise in the foreign language. The same holds for concept learning tasks, where students are learning a concept that is artifically devised, and as such, new to the student. Moreover, many of the studies in the sequencing activity types cluster were also teaching content to students for the first time. For example, Chi et al. [2009, 2010a,b] explicitly recruited students that had taken high school algebra but not college physics (which is what their dialogue-based ITS covered). Zhou et al. [2017], Shen and Chi [2016b], and Shen et al. [2018a,b] all ran experiments in a university course on discrete mathematics, where the ITS was actually used to teach course content to the students. This could also possibly explain why many of these studies found an aptitude-treatment interaction in favor of low-performing students: students who have more room to improve can benefit more from a better instructional policy than students who have more prior knowledge. On the other hand, almost all of the studies in the sequencing interdependent content cluster were on basic K-12 math skills, where the student was also presumably learning the content outside of using the systems in the studies. The only exceptions to this were the experiments run by Green et al. [2011], which actually showed that RL-induced policies did outperform random policies but not expert hand-crafted or heuristic baselines.

Baselines

Another factor that might affect why some studies were more likely to obtain significant results could be the choice of baseline policies. Among the 24 studies that found a significant effect or aptitude-treatment interaction, 17 of them (71%) only compared adaptive RL-induced policies to a random baseline policy and/or other RL-induced policies that have not been shown to perform well, rather than comparing to state-of-the-art baselines. On the other hand, among the studies that did not find a significant effect, only 6 of them (35%) only compared to random or RL-induced baseline policies. Of course, it is important to note that using such baselines is not always unreasonable. In some cases, researchers have compared to policies intentionally designed to perform poorly (e.g., by *minimizing* rewards according to a MDP), in order to determine if instructional sequencing has any effect on student learning whatsoever [Chi et al., 2010b, Geana, 2015, Lin et al., 2015]. In other cases, a random baseline may actually be a fairly decent policy. For instance, in cases where the policy must decide whether to assign worked examples or problem solving tasks, both actions have been shown to be beneficial in general, and hence a policy that sequences them randomly is thought to be reasonable [Shen et al., 2018a, Zhou et al., 2017]. Moreover, in paired association tasks, random policies may be reasonable because they happen to space problems fairly evenly. However, given that we now have better heuristics for potentially sequencing both worked-examples and problem solving tasks [Kalyuga and Sweller, 2005, Kalyuga et al., 2003] as well as paired-association tasks [Lindsey et al., 2014b, Pavlik and Anderson, 2008], it would be worthwhile to compare RL-induced policies to more advanced baselines (including other RL-induced policies that have been established to perform well) in future work for these domains.

Robust Evaluations

Finally, the use of robust evaluations, as discussed in the previous chapter, can help maximize the chance of finding successful instructional policies. Several of the studies that have been successful in using RL performed some kind of robust evaluation. Lindsey et al. [2014b] justified their use of a greedy heuristic policy by some simulations they ran (in a slightly different context) that showed the heuristic policy can be approximately as good as the optimal policy according to two different cognitive models (ACT-R and MCM). This can be thought of as a use case of a robust evaluation matrix to inform policy selection. In this case, finding a good heuristic policy was important because solving for the optimal policy would have been intractable. As discussed in Chapter 8, Rafferty et al. [2016a] also ran simulations that were analogous to using a robust evaluation matrix. Although they actually tested all policies that they ran in their simulations on actual students (for a better understanding of how effective various models and policies are), the kind of robust evaluation they did could have informed which policy to use if they did not want to test all policies [Doroudi et al., 2017a]. Mandel et al. [2014b] used importance sampling to choose a policy to run in their experiment. While importance sampling is impractical when considering many sequential decisions, Mandel et al. [2014b] only considered sequencing six levels in an educational game. On the other hand, several of the studies that did not show a significant difference between adaptive policies and baseline policies, including one of our own, only used a single model to simulate how well the policies would do, which model overestimated the performance of the adaptive policy [Chi et al., 2010a, Doroudi et al., 2017a, Rowe et al., 2014].

9.3.4 Summary

In short, it appears that reinforcement learning has yielded more benefits to students when one or more of the following things held:

- the sequencing problem was constrained in one or more ways (e.g., simple learning task or limited number of actions),
- statistical models of student learning were inspired by psychological theory,
- principles from psychology or the learning sciences suggested the importance of sequencing in that setting,
- students had fairly little prior knowledge coming in (but enough prior knowledge such that they could learn from the software they were interacting with),
- RL-induced policies were compared to relatively weak baselines (such as randomly presenting actions or policies that were not expected to perform well), and
- policies were tested in more robust and principled ways before being deployed on students.

This gives us a sense of the various factors that may influence the success of RL in instructional sequencing. Some of these factors suggest best practices which we believe might lead to more successfully using RL in future work. Others suggest practices that are actually best to avoid—such as using weak baseline policies when stronger baselines are available—in order to truly determine if RL-induced policies are beneficial for students. We now turn to how we can leverage some of these best practices in future work.

9.4 Planning for the Future

Our review of the empirical literature suggests that one exciting potential direction is to further combine data-driven approaches with psychological theories and principles from the learning sciences. Theories and principles can help guide (1) our choice of models, (2) the action space under consideration, and (3) our choice of policies. We briefly discuss the prospects of each of these in turn.

Psychological theory could help inform the use of reasonable models for particular domains as has been done in the case of paired-association tasks and concept learning tasks in the literature. These models can then be learned and optimized using data-driven RL techniques. Moreover, researchers should consider how psychological models can be developed for educationally relevant domains beyond just paired-association and concept learning tasks. Indeed such efforts could hopefully be productive both in terms of improving student learning outcomes in particular settings, as well as in testing and contextualizing existing or newly-developed theories.

Our results also suggest focusing on settings where the set of actions is restricted but still meaningful. For example, several of the studies described above consider the problem of sequencing worked examples and problem solving tasks, which meaningfully restricts the decision problem to two actions in an area where we know the sequence of tasks makes a difference [Kalyuga et al., 2003].

Finally, learning sciences principles can potentially help constrain the space of policies as well. For example, given that the expertise-reversal effect suggests that worked examples should precede problem solving tasks and that it is best to slowly fade away worked example steps over time, one could consider using RL to search over the space of policies that follow such a structure. This could mean rather than deciding at each time step what activity type to give to the student, the agent would simply need to decide when to switch to the next activity type. The expertise-reversal effect also suggests such switches should be based on the cognitive load on the student, which in turn can guide the representation used for the state space. Such policies have been implemented in a heuristic fashion in the literature on faded worked examples [Kalyuga and Sweller, 2005, Najar et al., 2016, Salden et al., 2010a], but researchers have not yet explored using RL to automatically find policies in this constrained space. Related to this, the learning sciences literature could suggest stronger baseline policies with which to compare RL-induced policies, as discussed in Section 9.3.3.

As the psychology and learning sciences literature identify more principles and theories of sequencing, such ideas can be integrated with data-driven approaches to guide the use of RL in instructional sequencing. Given that deep reinforcement learning has been gaining lots of traction in the past few years and will likely be increasingly applied to the problem of instructional sequencing, it seems especially important to find new ways of meaningfully constraining these approaches with psychological theory and learning sciences principles. A similar argument was made by Lindsey and Mozer [2016] when discussing their successful attempts of using a data-driven psychological model for instructional sequencing: “despite the power of big data, psychological theory provides essential constraints on models, and . . . despite the success of psychological theory in providing a qualitative understanding of phenomena, big data enables quantitative, individualized predictions of learning and performance."[8] Collaborations between learning scientists and machine learning researchers can help in finding effective ways of combining data and theory towards more impactful instructional sequencing.

However, given that finding a single plausible psychological model might be difficult in more complex settings, a complementary approach is to explicitly reason about robustness with respect to the choice of the model. Of course, such robust evaluations are not silver bullets, and they can be incorrect. However, even if the results do not match the predictions, this can help prompt new research directions in understanding the limitations of the models and/or instructional policies used. This happened with our second experiment (see Section 8.5.1), where comparing the experimental results with our simulations suggested the need for considering more nuanced temporal models of student learning.

Beyond these promising directions and suggestions, we note that the vast majority of the work we have reviewed consists of *system-controlled* methods of sequencing instruction that target *cognitive* changes. However, for data-driven instructional sequencing to have impact, we may need to consider broader ways of using instructional sequencing. The following are meant to be thought-provoking suggestions for consideration that build on current lines of research in the artificial intelligence in education community. In line with our recommendation to combine theory-driven and data-driven approaches, a common theme in many of these ideas is to combine machine intelligence with human intelligence, whether in the form of psychological theories, student choice, or teacher input.

9.4.1 Learner Control

In this review, we have only considered approaches where an automated instructional policy determines all decisions about what a learner should do, but sometimes the learners might know better what's best for themselves. Moreover, allowing for student choice could make students more motivated to engage with an instructional system [Fry, 1972, Kinzie and Sullivan, 1989]. Among the studies reported in our empirical review, only Atkinson [1972b] compared an RL-induced policy to a fully learner-controlled policy, and he found that while the learner-controlled policy was 53% better than random, it was not as good as the RL-induced policy (108% better than random). While this result was taken in favor of system-controlled policies, Atkinson [1972b] suggested that while the learner should not have complete control over the sequencing of activities, there is still "a place for the learner's judgments in making instructional decisions." There are a number of ways in which a machine's instructional decisions could be combined with student choice. One is for the agent to make recommendations about what actions the student should take, but ultimately leave the choice up to the student. This type of shared control has been shown to succesfully improve learning beyond system control in some settings [Corbalan

[8] Despite the fact that deep learning techniques are high variance, they have been successful in a variety of applications. Thus, one might wonder whether deep RL with big data is sufficient for finding good instructional policies, without the need for theory. We will revisit this question in the next chapter.

et al., 2008, Long and Aleven, 2016]. Green et al. [2011] found that expert policies do better than random policies, regardless of whether either policy made all decisions or gave the student a choice of three actions to take. Cumming and Self [1991] also describe such a form of shared control in their vision of "intelligent educational systems," where the system is a collaborator to the student rather than an instructor. Another approach would be for the agent to make decisions where it is confident its action will help the student, and leave decisions that it is less confident about up to the student. For example, an instructional policy could give a student a remedial problem when the student seems to be struggling with some basic skills, but otherwise let the student choose for themself. Finally, RL-induced policies could take learner decisions and judgements as inputs to consider during decision making (e.g., as part of the state space). For instance, Nelson et al. [1994] showed that learners can effectively make judgments of learning (JOLs) in paired-association tasks, and remarked that JOLs could be used by MDPs to make instructional decisions for students. Such a form of shared control has recently been considered in the RL framework for robotics applications [Javdani et al., 2018, Reddy et al., 2018], but has not been considered in the context of instructional sequencing to our knowledge.

9.4.2 Teacher Control

Building on the previous point, sometimes when an instructional policy does not know what to do, it could inform the teacher and have the teacher give guidance to the student. For example, Beck and Gong [2013] have shown that mastery learning policies could lead to "wheel-spinning" where students cannot learn a particular skill, perhaps because the policy cannot give problems that help the student learn. Detectors have been designed to detect when students are wheel-spinning [Gong and Beck, 2015, Matsuda et al., 2016]. These detectors could then relay information back to teachers, for example through a teacher dashboard [Aleven et al., 2016b] or augmented reality analytics software [Holstein et al., 2018], so that teachers know to intervene. In these cases, an RL agent could encourage the teacher to pick the best activity for the student to work on (or a recommend a set of activities that the student could choose from). In some cases, this will likely require the teacher giving an explanation or addressing a student's misconception outside of the software. The teacher's input could also potentially help the agent learn how to make better decisions for that particular student in the future. Finding the right balance between learner-control, teacher-control, and system-control is an open and important area of research in instructional sequencing.

9.4.3 Beyond the Cognitive

Almost all of the empirical studies we have reviewed used cognitive models of learning that were designed to lead to cognitive improvements in learning (e.g., how much students learned or how fast they learned). However, RL could also take into account affective, motivational, and metacognitive features in the state space and could also be used in interventions that target these non-cognitive aspects of student learning (i.e., by incorporating them in reward functions). For example, could a policy be derived to help students develop a growth mindset or to help students develop stronger metacognitive abilities? While detecting affective states is a growing area of research in educational data mining and AIED [Baker et al., 2012, Calvo and D'Mello, 2010], only a few studies have considered using affective states and motivational features to adaptively sequence activities for students [Aleven et al., 2016a]. For example, Baker et al. [2006] used a detector that predicts when a student is gaming the system in order to assign students supplementary exercises when they exhibit gaming behavior and Mazziotti et al. [2015] used measures of both the student's cognitive state and affective state to determine the next activity to give the student. There has also been work on adaptive learning technologies that improve students' self-regulatory behaviors, but this has not been studied in the context of instructional sequencing [Aleven et al., 2016a]. While there is a risk that modeling metacognition or affect may be even harder than modeling students' cognitive states in a reinforcement learning framework, there may be certain places where we can do so effectively, and the impact of such interventions might be larger than solely cognitive interventions.

9.4.4 Conclusion

We have discussed a number of ways in which human intelligence can be combined with machine intelligence towards potentially improving instructional sequencing. Psychological theories and the learning sciences can guide the process of inducing instructional policies before applying data-driven methods. This could also include considering theories of motivation and affect and how they might impact the optimal sequence of instructional activities. Moreover, RL methods could consider multiple competing theories of how students learn in order to make more robust decisions. Finally, the RL agent could also include the student and teacher in the decision-making process to ultimately find the instructional activity that is best suited to each student at any given point in time. Only time can tell if following these practices will actually lead to more impactful instructional sequencing, but we hope that this review helps us carve the most productive path forward.

However, it is important to note that the process of using reinforcement learning for instructional sequencing has been beneficial beyond its impact on student learning. Perhaps the biggest success of framing instructional sequencing as a reinforcement learning problem has actually been its impact on the fields of artificial intelligence, operations research, and student modeling. As mentioned in our historical review, investigations in optimizing instruction have led to the formal development of partially observable Markov decision processes [Smallwood and Sondik, 1973, Sondik, 1971], which is now an important area of study in operations research and artificial intelligence. More recently, in some of our own work, the challenge of estimating the performance of different instructional policies has led to advancements in general statistical estimation techniques [Doroudi et al., 2017c, Mandel et al., 2014b] that are relevant to treatment estimation in healthcare, advertisement selection, and many other areas. Finally, in the area of student modeling, our robust evaluation matrix can help researchers not only find good policies but also discover the limitations of the models when a policy under-delivers. In the words of Atkinson, "the development of a theory of instruction cannot progress if one holds the view that a complete theory of learning is a prerequisite. Rather, advances in learning theory will affect the development of a theory of instruction, and conversely the development of a theory of instruction will influence research on learning” [Atkinson, 1972a]. Thus, not only should we use theories of learning to improve instructional sequencing, but also by trying to improve instructional sequencing, perhaps we can gain new insights about how people learn.

Part III

Conclusion

Chapter 10

Integrating Human and Machine Intelligence

> I should warn you, in conclusion, to beware of the likes of us. We do not have a tested theory of instruction to offer you. What is quite plain is that one is needed and I would propose that we work together in its forging.
>
> Jerome Bruner, 1963

In this thesis, I have proposed a number of methods to tackle various aspects of semi-automated curriculum design that combine machine learning, human computation, and principles from the learning sciences. Due to the interdisciplinary nature of this work, my thesis makes contributions to a number of fields. Here I enumerate some of the key contributions to each field, which also forms a summary of my entire thesis:

- **Learning Sciences**: In Part I, I discussed various experimental results for how to leverage peer work to benefit other learners. While the learning sciences have an extensive literature on the benefit of worked examples and active over passive learning, these results are typically in the context of learners interacting with expert-designed content. My experiments have extended some of these results to the context of interacting with learner-generated work:
    - For web search tasks (Chapter 3), seeing two expert examples led to significantly higher worker accuracy on future tasks than not receiving any training (Glass' Δ = 0.42). Moreover, validating worker solutions seemed to be effective, provided that the solutions were sufficiently long (regardless of whether they were correct or not). In particular, workers who validated solutions that were long enough[1], performed considerably better than workers who saw expert examples (Glass' Δ = 0.55). Although this result was not confirmed with a proper experiment, it suggests that validating

[1] 500-800 characters for the first solution, and more than 1000 solutions for the second solution

peer work can be an effective form of training, if solutions are appropriately curated (e.g., by length).

  - For product comparison review tasks (Chapter 4), seeing random peer-generated examples was no better than receiving no training, and seeing a single randomly chosen peer example may have even decreased performance. However, seeing two high quality examples led to significantly higher accuracy on one of three test tasks (Glass' $\Delta = 0.4$) and trended to be better on the remaining test tasks. This result suggests that simply reading peer examples can be an effective form of training provided that the examples are of sufficiently high quality. Moreover, we found that among high quality examples, some seemed to result in better future performance than others, suggesting the need for more sophisticated content curation than relying on quality (as rated by workers).

- **Crowdsourcing / Human Computation**: All of my work on learnersourced content generation was in the context of training crowdworkers to do complex tasks. Therefore, the results described above also have implications for how to effectively train crowdworkers. This has implications for crowdsourcing requesters who want tasks done effectively, but more importantly (to me), this has implications for how to help crowdworkers obtain new skills and opportunities to do more complex work. Ultimately, the vision of this work is to develop semi-automated pipelines to help crowdworkers obtain new skills, which also has implications for retraining workers to handle the ever-changing demands of the future of work.

- **Educational Data Mining (EDM) / Artificial Intelligence in Education (AIED)**: Much of the contribution of my work in Part II was bringing over key concepts from machine learning and statistics, such as model misspecification, model robustness, and the bias-variance tradeoff, to better classify approaches to instructional sequencing and understanding their limitations. Specifically, the key contributions were as follows:

  - In Chapter 6, I discussed how semantically degenerate parameters for the Bayesian knowledge tracing model cannot be due to model identifiability, as was previously believed [Beck and Chang, 2007], because BKT is an identifiable model [see Doroudi and Brunskill, 2017]. Instead, I proposed that semantically degenerate parameters could arise from statistical model misspecification, specifically when student learning is actually more incremental than BKT suggests. Moreover, I showed that this kind of model misspecification could lead to mistakenly assuming students have mastered skills when they have not.

  - In Chapter 7, I analyzed the equitability of mastery learning with BKT when there are students of different types (e.g., fast and slow learners). I showed that while mastery learning is more equitable than one-size-fits-all instruction, it can still be inequitable when (1) the BKT model is not individualized to different types of students, and (2) the assumptions of the BKT model are incorrect, specifically when student learning is more incremental than BKT suggests.

  - In Chapter 8, I proposed a new tool, the robust evaluation matrix (REM), for evaluating different instructional policies by leveraging multiple models of student learning. The tool was proposed to mitigate making decisions based on a single biased (i.e., misspecified) student model (such as BKT), while simultaneously avoiding policy evaluation techniques that are too high variance to be practical. Through case studies, I showed how REM could be used to avoid deploying ineffective instructional policies, as well as to identify robustly effective policies.

- In Chapter 9, I reviewed the empirical literature on using reinforcement learning for instructional sequencing. One of the key findings of this review was that the most successful prior approaches seem to have been ones that leveraged psychological theories (e.g., in informing the choice of student models). In contrast, I found that more purely data-driven approaches to instructional sequencing have had very little empirical success.

While I hope each of these individual contributions provide new insights to their respective fields, the broader theme of my dissertation has been demonstrating various ways in which human intelligence and machine intelligence can be combined to improve semi-automated curriculum design. Human intelligence can come in a number of forms. It can come in the form of theories and principles in the learning sciences and psychology. For example, the robust evaluation matrix relies on using multiple models of student learning that can be motivated by different (possibly competing) theories of learning, but ultimately fit to data. However, REM requires human intelligence beyond the models that it uses. I purposely framed REM as a tool that guides researchers and practitioners in deciding which policy to use (if any), as opposed to an algorithm that tells someone what policy to deploy. Moreover, REM will likely be used in an iterative fashion in most applications, as was the case, when we used it for our second experiment on our fractions ITS. One may start with an initial set of policies and models, but after using REM, one may find limitations in these models and/or policies, which could lead to investigations that require (1) improving the existing models perhaps to conform better to some theory of learning, (2) bringing in new models, (3) fitting existing models to different sub-populations of students, and/or (4) searching for new policies. As is implicitly the case with a lot of data science pipelines, REM is a human-machine hybrid tool for developing instructional policies.

At the end of Chapter 9, I briefly speculated about other ways in which human intelligence can inform instructional sequencing. One of these is to consider who should be in control of the sequence of instruction: the learner, the teacher, or an algorithm? Naturally, I believe a hybrid approach will be optimal in most cases, but more work is needed in determining how to combine these different inputs to ultimately adaptively guide the curriculum each learner experiences. Some recent work has looked into preliminary steps in considering how to design hybrid teaching systems that involve judgements made by teachers and AI [Holstein, 2018, Holstein et al., 2018] as well as students [Molenaar et al., 2019].

Although I did not look at using learner control in the sequencing of content, the first part of my thesis explored using learner contributions in the creation of content. This content then needs to be given to learners in ways that are productive, which can be guided by psychological theories (e.g., the benefit of active engagement, the benefit of worked examples etc.). But this content also needs to be curated, which can be guided by machine learning algorithms. The machine may discover various principles that makes up effective learner-generated content, which can in turn inform learning science theories and lead to future experiments that explore the design of effective content.

The approaches to content creation and instructional sequencing that I have explored in my thesis are complementary. Ultimately, I envision approaches to semi-automated curriculum design that take input from teachers and learners in real-time to determine how to sequence activities that consist of expert-generated content and learner-generated content. Of course, such a fully-integrated approach to human-machine hybrid curriculum design may be a futuristic vision, but some seeds of such a vision are planted in this thesis.

The idea of combining human intelligence and machine intelligence is a broader theme in the history of artificial intelligence. Early pioneers in AI aimed to build artificial intelligence that aimed to mimic human intelligence. Early approaches by Herbert Simon and Allen Newell at Carnegie Mellon relied on capturing human expertise and developing computer programs that could replicate how humans solve problems. Simon and Newell were fundamentally cognitive scientists who were interested in understanding how humans and machines can think, reason, and learn. Their work led to the development of intelligent tutoring systems that relied on production rules and symbolic information processing theories. This approach persists to today as a significant thrust of research on AI in education. Around the same time as Newell and Simon, other AI pioneers such as Marvin Minsky and Seymour Papert were also interested in how humans and machines learn, although their approach was primarily focused on learning in children rather than problem-solving in expert adults. Naturally, their interest in how children learn was fundamentally related to education as well, and their work has been very influential in the formation and growth of the field of the learning sciences. The field of the learning sciences was largely founded (and named) by Roger Schank, another pioneer in the early days of AI who was interested in how people learn by examples and stories. However, despite the influence of AI pioneers on educational technology and the learning sciences, these early approaches to AI have largely been replaced by machine learning, especially deep learning, in recent years. AI has shifted from being theory-driven to being data-driven, from emphasizing human intelligence to emphasizing machine intelligence.

Deep learning is gaining a lot of popularity because of its success in a variety of application areas, such as computer vision, natural language processing, and game playing. In Chapter 9, we saw that deep learning is also becoming a popular technique in automated instructional sequencing. It is worth revisiting the question of whether deep learning is sufficient to make broad advances in enhancing curriculum design. Theoretically, deep neural networks are known to be very high variance algorithms. For example, it is well known that they are able to model arbitrary functions (albeit with prohibitively large networks) [Hornik, 1991]. However, recent research has shown that for various reasons that are still under investigation, they often do generalize well in practice with enough data [Lin et al., 2017, Zhang et al., 2016]. Therefore, it might be tempting to think that deep learning can greatly enhance instructional sequencing, just as it has made great advances in other applications. However, I believe learning is *fundamentally* different from images, language, and games. In some sense, developing an accurate computational model of human cognition and learning would require developing artificial general intelligence itself (whereas developing a computational model that can solve a specific type of game requires only a very specialized AI technology). Moreover, so long as we use probabilistic models to model learning, even under the best such model, there are limits to how accurately we can predict when a student has learned a skill, because we can only observe knowledge through noisy observations [Beck and Xiong, 2013, Chen et al., 2018]. I thus find it unlikely that simply more data and more advanced algorithms is all we need to tackle curriculum design. I do not mean to claim that deep learning have no place in educational applications, but rather that I believe it should be combined with theory-driven approaches and other forms of human intelligence.

Moreover, despite the recent popularity of deep learning, throughout the history of AI, a number of researchers have advocated for integrating the theory-driven and logic-based approaches rooted in human cognition with data-driven machine learning approaches [Bach et al., 2015, Domingos et al., 2006, Hu et al., 2016, Minsky and Papert, 1969]. For example, despite being an early critic of connectionist neural network approaches [Minsky and Papert, 1969], in an article called "Logical versus Analogical or Symbolic versus Connectionist or Neat versus Scruffy," Minsky [1991] admitted that symbolic approaches and connectionism have both made advances, but to create truly intelligent systems, both approaches need to be integrated:

> neither purely connectionist nor purely symbolic systems seem to be able to support the sorts of intellectual performances we take for granted even in young children...I'll argue that the solution lies somewhere between these two extremes, and our problem will be to find out how to build a suitable bridge.

In an early report on their work in AI, Minsky and Papert [1971] explicitly state limitations of research on Newell and Simon's early work on "Automatic Theorem Provers" such as the lack of emphasis on "a highly organized structure of especially appropriate facts, models, analogies, planning mechanisms, self-discipline procedures" as well as the lack of heuristics in solving proofs (e.g., mathematical insights used in solving the proof that are not part of the proof itself). They then use this to motivate the need for what they call "micro-worlds":

> We are dependent on having simple but highly developed models of many phenomena. Each model—or "micro-world" as we shall call it—is very schematic...we talk about a fairyland in which things are so simplified that almost every statement about them would be literally false if asserted about the real world. Nevertheless, we feel they are so important that we plan to assign a large portion of our effort to developing a collection of these micro-worlds and finding how to embed their suggestive and predictive powers in larger systems without being misled by their incompatibility with literal truth. We see this problem—of using schematic heuristic knowledge—as a central problem in Artificial Intelligence.

This approach is very much reminiscent of the robust evaluation matrix. Minsky and Papert recognized that learning is inherently complex, so rather than finding a unified theory of cognition and learning (as motivated by Newell), they tried to find models that were admittedly wrong in many ways, but brought insights on how people learn in a small way. By connecting these micro-worlds in a meaningful way, they hoped to create a more bottom-up theory of how people learn. Here, I proposed using various models (or micro-worlds) to simulate different instructional policies. The richer and more multi-faceted the models, the more robust predictions we can make about instructional techniques. But this general idea of developing many micro-worlds, each of which give a glimpse of the learning process can have wider-ranging consequences for the design of semi-automated curricula. The concept of a microworld has since become very important in Papert's educational theories [Papert, 1980, 1987][2], but I believe its role in early AI is largely forgotten.

Perhaps, by turning to the history of AI and prior approaches towards integrating human and machine intelligence we can find richer ways of understanding how people learn and how to construct meaningful curricula for these lifelong learners. This in turn might lead to new insights on how to integrate human and machine intelligence that can contribute to the broader AI literature. Although I have only presented a few simple steps towards integrated semi-automated curriculum design, I hope that these ideas will lead to more ways in which we can better understand and enhance learners' formal and informal educational experiences.

[2] In fact, Papert believed microworlds that learners could interact with on computers would replace the need for curricula altogether: "At some future time, complex networks of microworlds that touch on many sectors of knowledge will be the staple diet of learning, and will replace the present concept of 'curriculum.'"

Appendix

Appendix A

Web Search Task Experimental Materials

A.1 Web Search Questions

The following are the web search questions that we had crowdworkers complete The answers are hidden to avoid spoilers for any readers interested in attempting these web search tasks. The answers can be obtained by highlighting and pasting the text into a text editor.

Training Questions

- **Question X**: Living by the Cam with around 120,000 other people, what do your fellow countrymen refer to you and your neighbors by? (Hint: It's 13 letters in plural.)
  Answer:

- **Question Y**: The Plaster Cramp is the title of a fictional book in the fictional Library of Babel as envisioned by Jorge Luis Borges. There is another book in this library whose name only has a meaning in a fictional language in one of Borges' other short stories. The name of this other book (in the fictional language) has to do with what celestial object?
  Answer:

Test Questions

- **Question A**: The number one producer of pistachios in the world (at least until a few years ago) is also the number one producer of what plant stigma?
  Answer:

- **Question B**: What is the name of the closest freeway to the Happiest Place on Earth? (We're looking for its three word name in that region–not the Interstate number!)
  Answer:

- **Question C**: If you wanted to travel by ferry to a country where many natives believe in elves, in what city would you land?

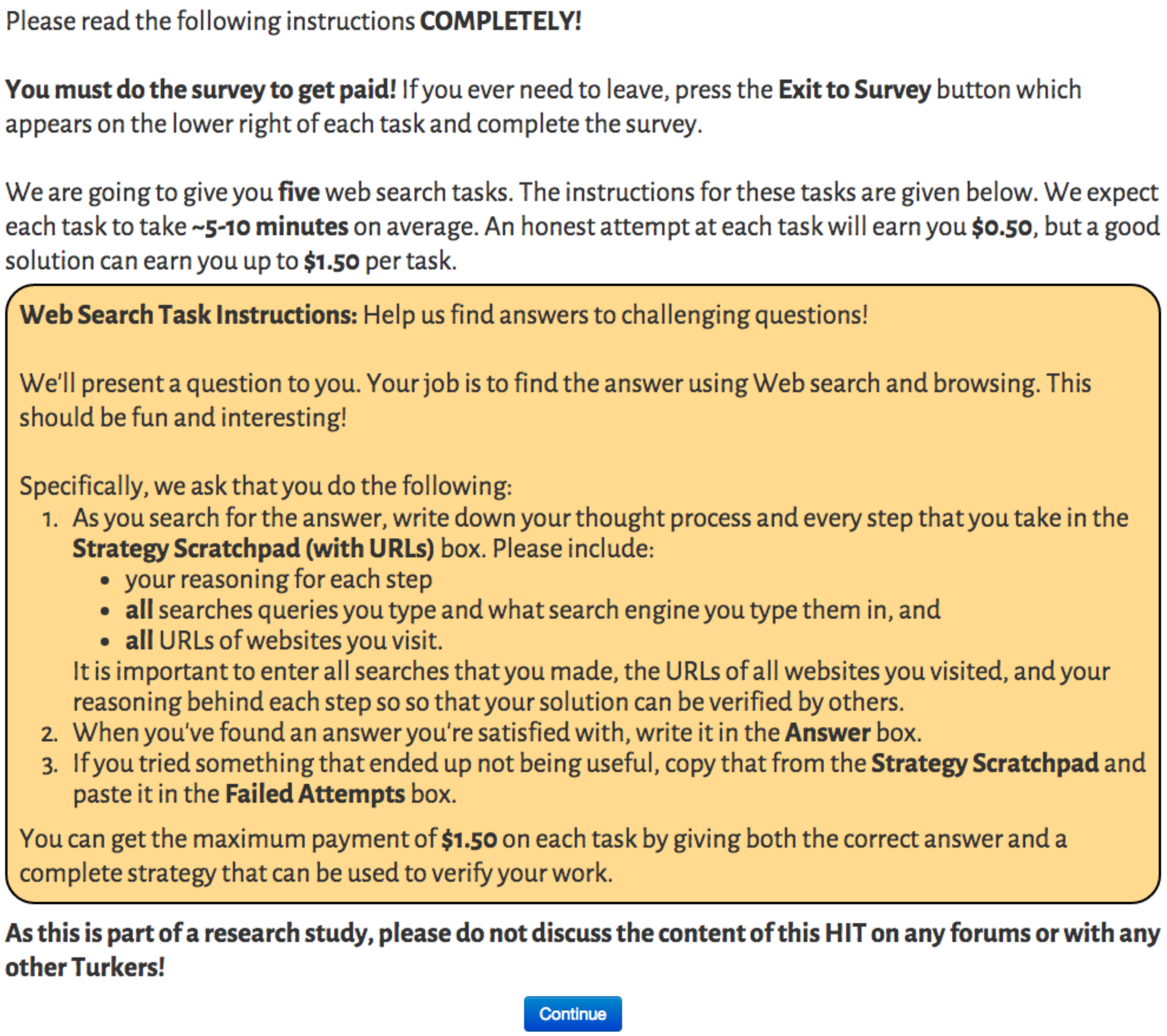
Answer:

- **Question D**: What facility is jointly operated by (1) a government organization headquartered in Washington D.C., which also operates nine other similar facilities, and (2) a university that was responsible for changing the Hollywood sign in 1987?
Answer:

- **Question E**: One founder of calculus—not the one with the apple—had the idea of a universal language of sorts. What twentieth-century mathematician later formed an obsession around this idea?
Answer:

A.2 Task Instructions and Surveys

The following figures show images of instructions and surveys presented to workers for three of the five conditions (control, solution, and validation) in Experiment I. The remaining instructions and surveys follow similar templates.


Please read the following instructions **COMPLETELY!**

You must do the survey to get paid! If you ever need to leave, press the **Exit to Survey** button which appears on the lower right of each task and complete the survey.

We are going to give you **five** web search tasks. The instructions for these tasks are given below. We expect each task to take **~5-10 minutes** on average. An honest attempt at each task will earn you **$0.50**, but a good solution can earn you up to **$1.50** per task.

Web Search Task Instructions: Help us find answers to challenging questions!

We'll present a question to you. Your job is to find the answer using Web search and browsing. This should be fun and interesting!

Specifically, we ask that you do the following:

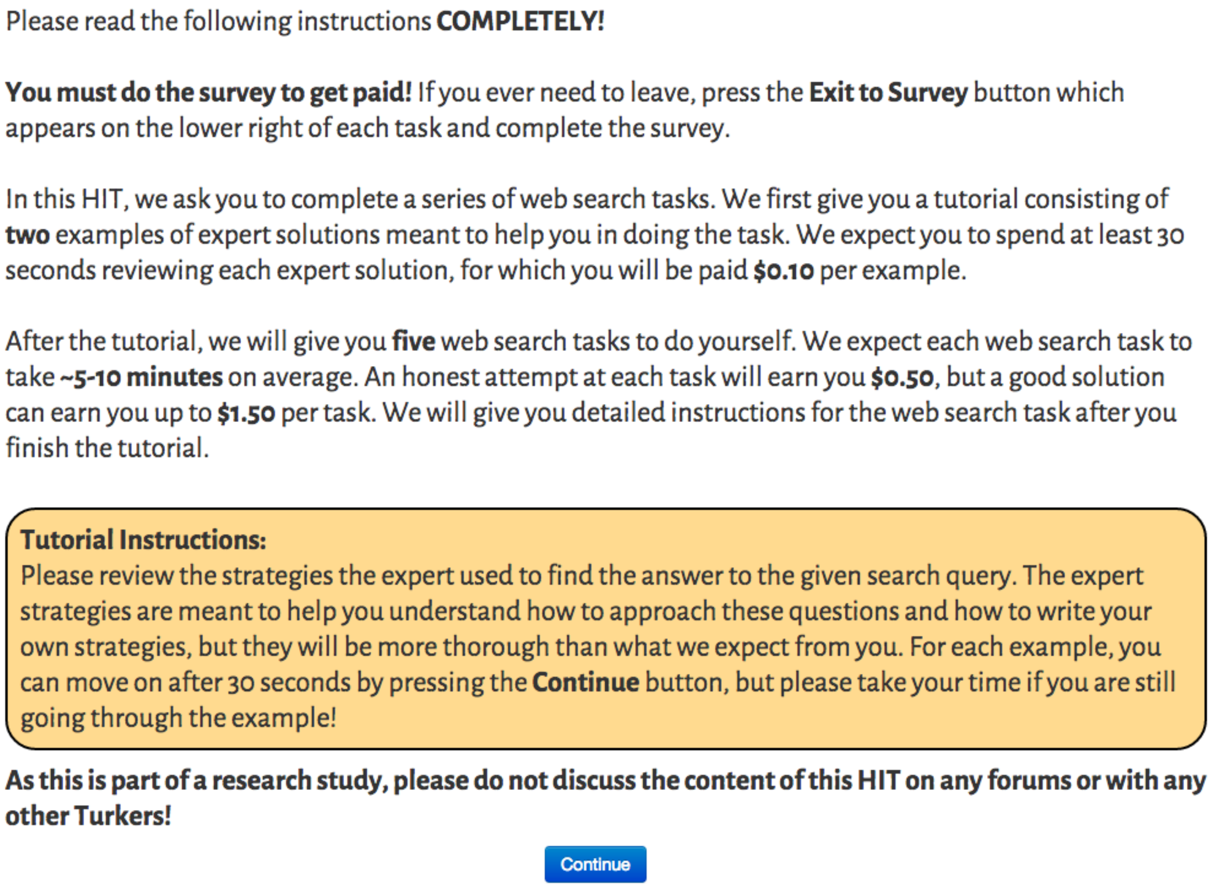
1. As you search for the answer, write down your thought process and every step that you take in the **Strategy Scratchpad (with URLs)** box. Please include:
   - your reasoning for each step
   - **all** searches queries you type and what search engine you type them in, and
   - **all** URLs of websites you visit.

   It is important to enter all searches that you made, the URLs of all websites you visited, and your reasoning behind each step so so that your solution can be verified by others.
2. When you've found an answer you're satisfied with, write it in the **Answer** box.
3. If you tried something that ended up not being useful, copy that from the **Strategy Scratchpad** and paste it in the **Failed Attempts** box.

You can get the maximum payment of **$1.50** on each task by giving both the correct answer and a complete strategy that can be used to verify your work.

As this is part of a research study, please do not discuss the content of this HIT on any forums or with any other Turkers!

Figure A.1: Control condition instructions



Please read the following instructions **COMPLETELY!**

You must do the survey to get paid! If you ever need to leave, press the **Exit to Survey** button which appears on the lower right of each task and complete the survey.

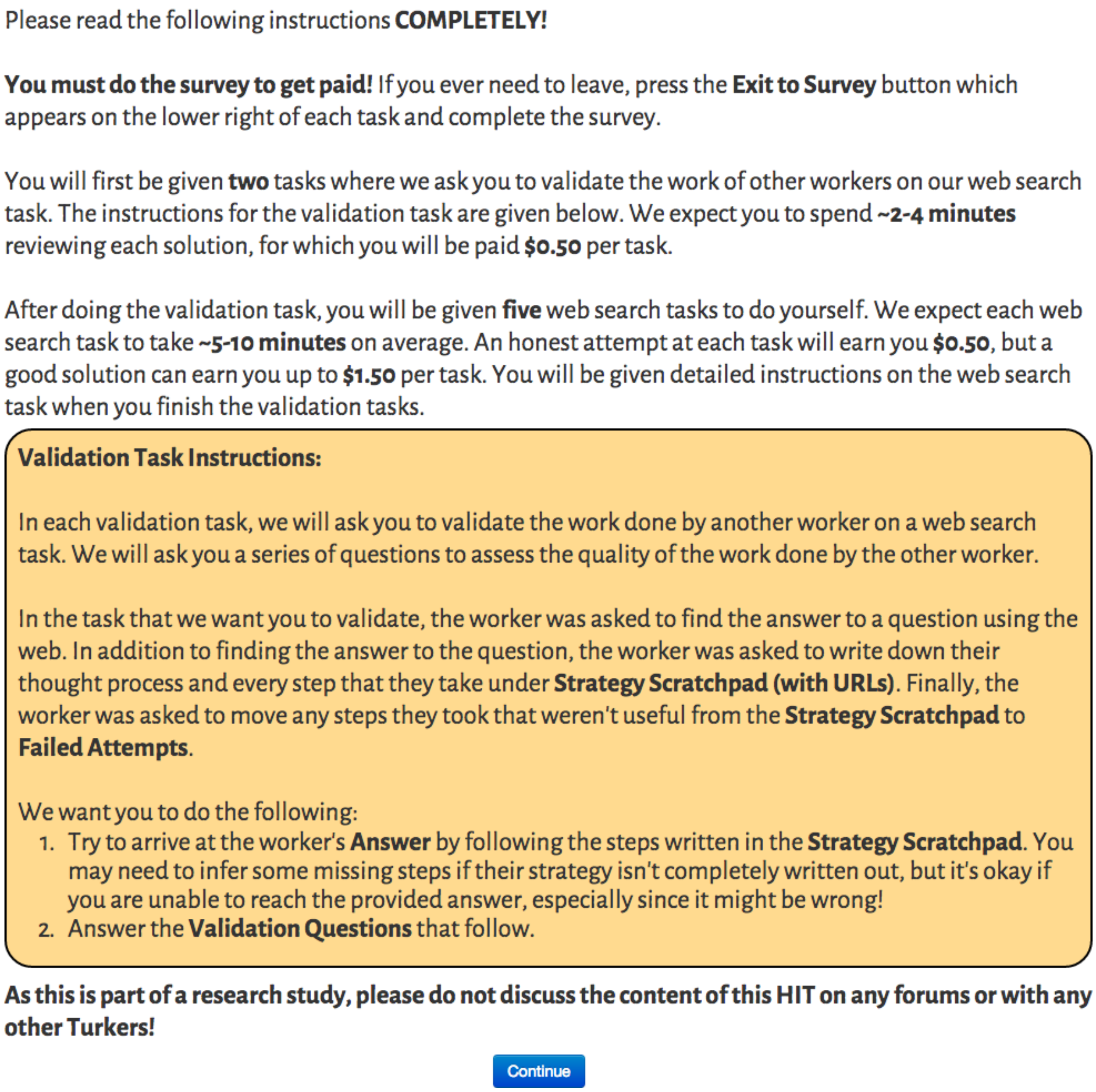
In this HIT, we ask you to complete a series of web search tasks. We first give you a tutorial consisting of **two** examples of expert solutions meant to help you in doing the task. We expect you to spend at least 30 seconds reviewing each expert solution, for which you will be paid **$0.10** per example.

After the tutorial, we will give you **five** web search tasks to do yourself. We expect each web search task to take **~5-10 minutes** on average. An honest attempt at each task will earn you **$0.50**, but a good solution can earn you up to **$1.50** per task. We will give you detailed instructions for the web search task after you finish the tutorial.

Tutorial Instructions:
Please review the strategies the expert used to find the answer to the given search query. The expert strategies are meant to help you understand how to approach these questions and how to write your own strategies, but they will be more thorough than what we expect from you. For each example, you can move on after 30 seconds by pressing the **Continue** button, but please take your time if you are still going through the example!

As this is part of a research study, please do not discuss the content of this HIT on any forums or with any other Turkers!

Figure A.2: Example condition instructions



Please read the following instructions **COMPLETELY!**

You must do the survey to get paid! If you ever need to leave, press the **Exit to Survey** button which appears on the lower right of each task and complete the survey.

You will first be given **two** tasks where we ask you to validate the work of other workers on our web search task. The instructions for the validation task are given below. We expect you to spend **~2-4 minutes** reviewing each solution, for which you will be paid **$0.50** per task.

After doing the validation task, you will be given **five** web search tasks to do yourself. We expect each web search task to take **~5-10 minutes** on average. An honest attempt at each task will earn you **$0.50**, but a good solution can earn you up to **$1.50** per task. You will be given detailed instructions on the web search task when you finish the validation tasks.

Validation Task Instructions:

In each validation task, we will ask you to validate the work done by another worker on a web search task. We will ask you a series of questions to assess the quality of the work done by the other worker.

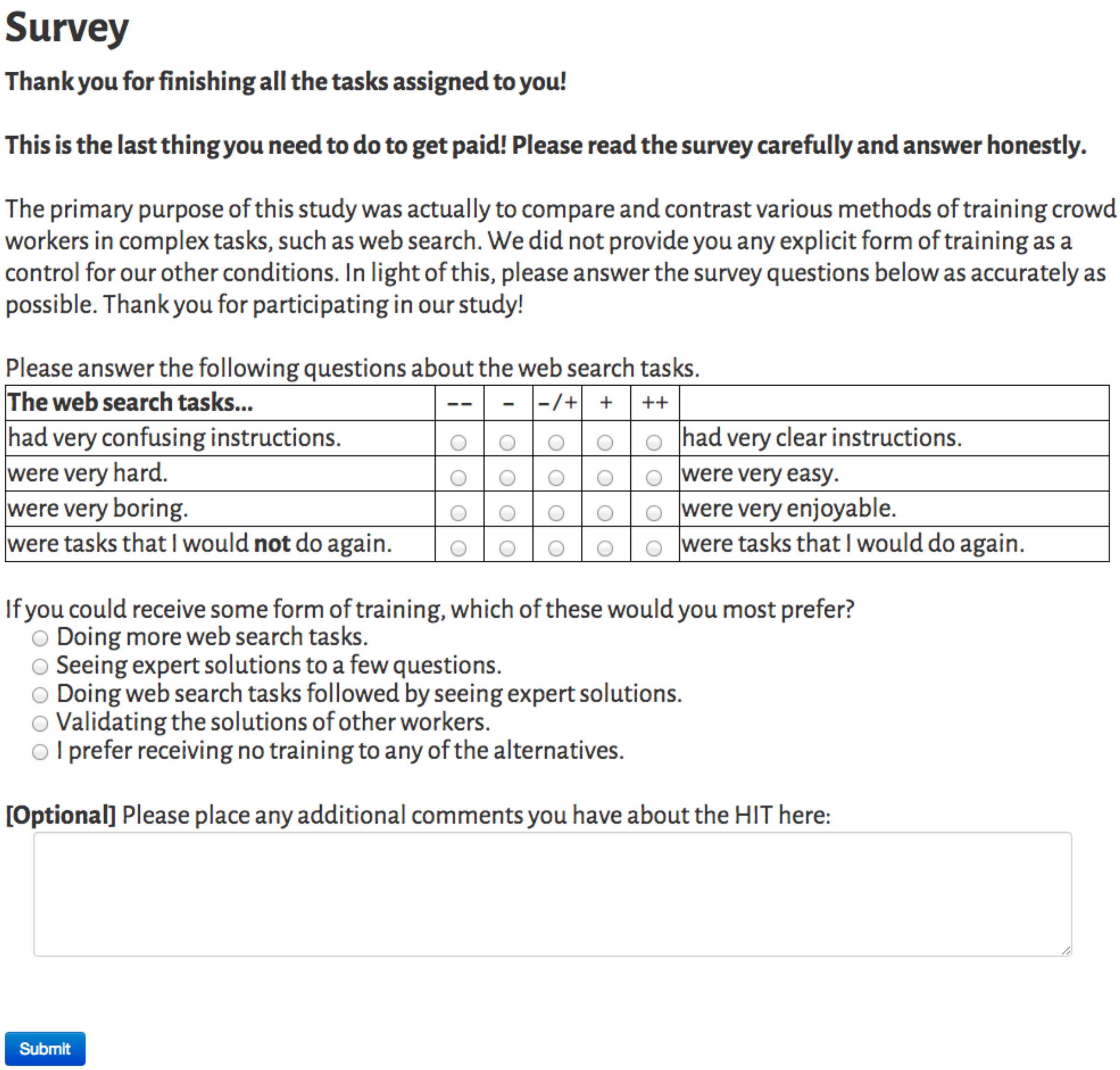
In the task that we want you to validate, the worker was asked to find the answer to a question using the web. In addition to finding the answer to the question, the worker was asked to write down their thought process and every step that they take under **Strategy Scratchpad (with URLs)**. Finally, the worker was asked to move any steps they took that weren't useful from the **Strategy Scratchpad** to **Failed Attempts**.

We want you to do the following:

1. Try to arrive at the worker's **Answer** by following the steps written in the **Strategy Scratchpad**. You may need to infer some missing steps if their strategy isn't completely written out, but it's okay if you are unable to reach the provided answer, especially since it might be wrong!
2. Answer the **Validation Questions** that follow.

As this is part of a research study, please do not discuss the content of this HIT on any forums or with any other Turkers!

Figure A.3: Validation condition instructions



Survey

Thank you for finishing all the tasks assigned to you!

This is the last thing you need to do to get paid! Please read the survey carefully and answer honestly.

The primary purpose of this study was actually to compare and contrast various methods of training crowd workers in complex tasks, such as web search. We did not provide you any explicit form of training as a control for our other conditions. In light of this, please answer the survey questions below as accurately as possible. Thank you for participating in our study!

Please answer the following questions about the web search tasks.

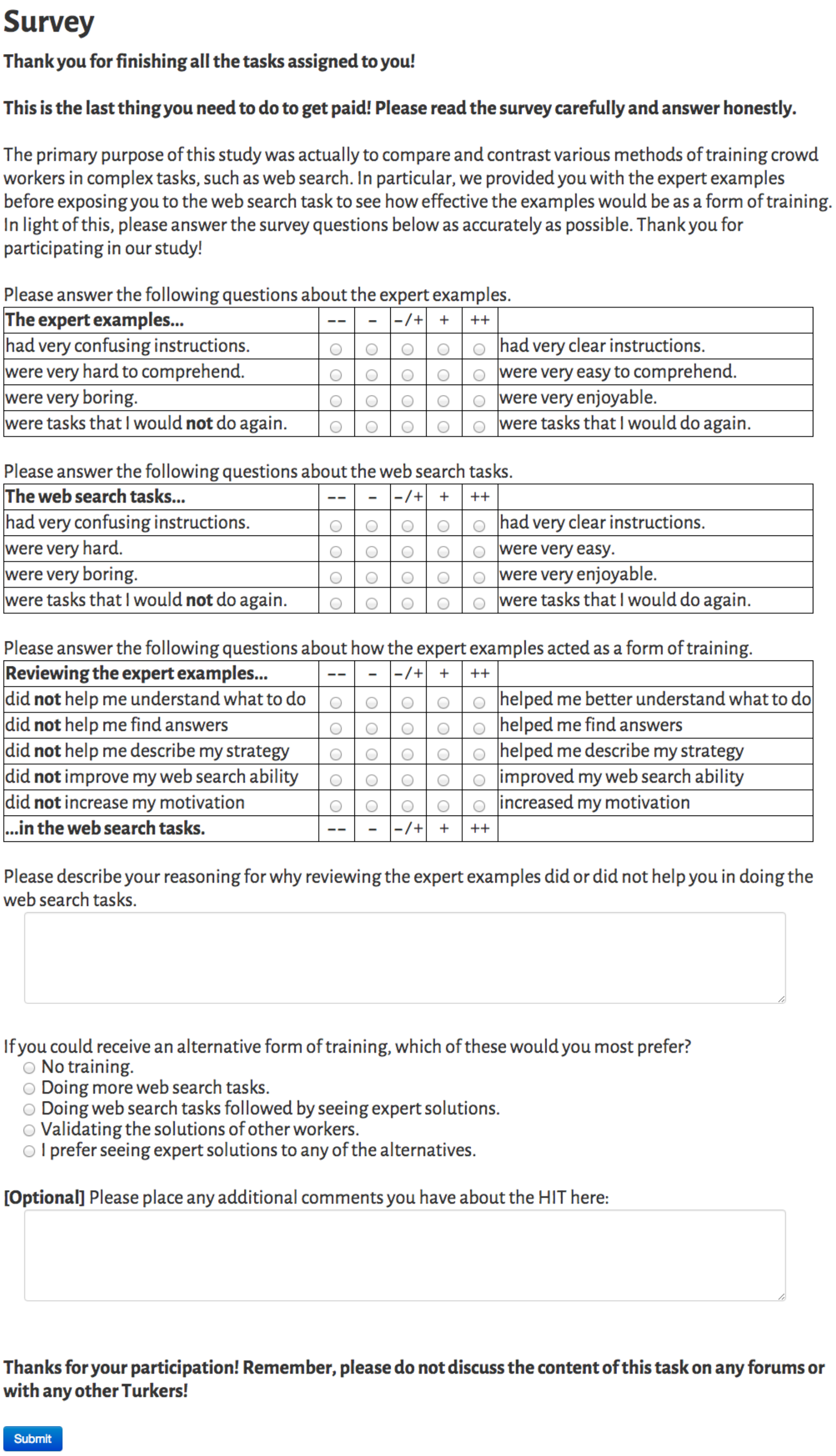
The web search tasks...	--	-	-/+	+	++	
had very confusing instructions.	○	○	○	○	○	had very clear instructions.
were very hard.	○	○	○	○	○	were very easy.
were very boring.	○	○	○	○	○	were very enjoyable.
were tasks that I would **not** do again.	○	○	○	○	○	were tasks that I would do again.

If you could receive some form of training, which of these would you most prefer?

○ Doing more web search tasks.
○ Seeing expert solutions to a few questions.
○ Doing web search tasks followed by seeing expert solutions.
○ Validating the solutions of other workers.
○ I prefer receiving no training to any of the alternatives.

[Optional] Please place any additional comments you have about the HIT here:

Figure A.4: Control condition survey



Survey

Thank you for finishing all the tasks assigned to you!

This is the last thing you need to do to get paid! Please read the survey carefully and answer honestly.

The primary purpose of this study was actually to compare and contrast various methods of training crowd workers in complex tasks, such as web search. In particular, we provided you with the expert examples before exposing you to the web search task to see how effective the examples would be as a form of training. In light of this, please answer the survey questions below as accurately as possible. Thank you for participating in our study!

Please answer the following questions about the expert examples.

The expert examples...	--	-	-/+	+	++	
had very confusing instructions.	○	○	○	○	○	had very clear instructions.
were very hard to comprehend.	○	○	○	○	○	were very easy to comprehend.
were very boring.	○	○	○	○	○	were very enjoyable.
were tasks that I would **not** do again.	○	○	○	○	○	were tasks that I would do again.

Please answer the following questions about the web search tasks.

The web search tasks...	--	-	-/+	+	++	
had very confusing instructions.	○	○	○	○	○	had very clear instructions.
were very hard.	○	○	○	○	○	were very easy.
were very boring.	○	○	○	○	○	were very enjoyable.
were tasks that I would **not** do again.	○	○	○	○	○	were tasks that I would do again.

Please answer the following questions about how the expert examples acted as a form of training.

Reviewing the expert examples...	--	-	-/+	+	++	
did **not** help me understand what to do	○	○	○	○	○	helped me better understand what to do
did **not** help me find answers	○	○	○	○	○	helped me find answers
did **not** help me describe my strategy	○	○	○	○	○	helped me describe my strategy
did **not** improve my web search ability	○	○	○	○	○	improved my web search ability
did **not** increase my motivation	○	○	○	○	○	increased my motivation
...in the web search tasks.	--	-	-/+	+	++	

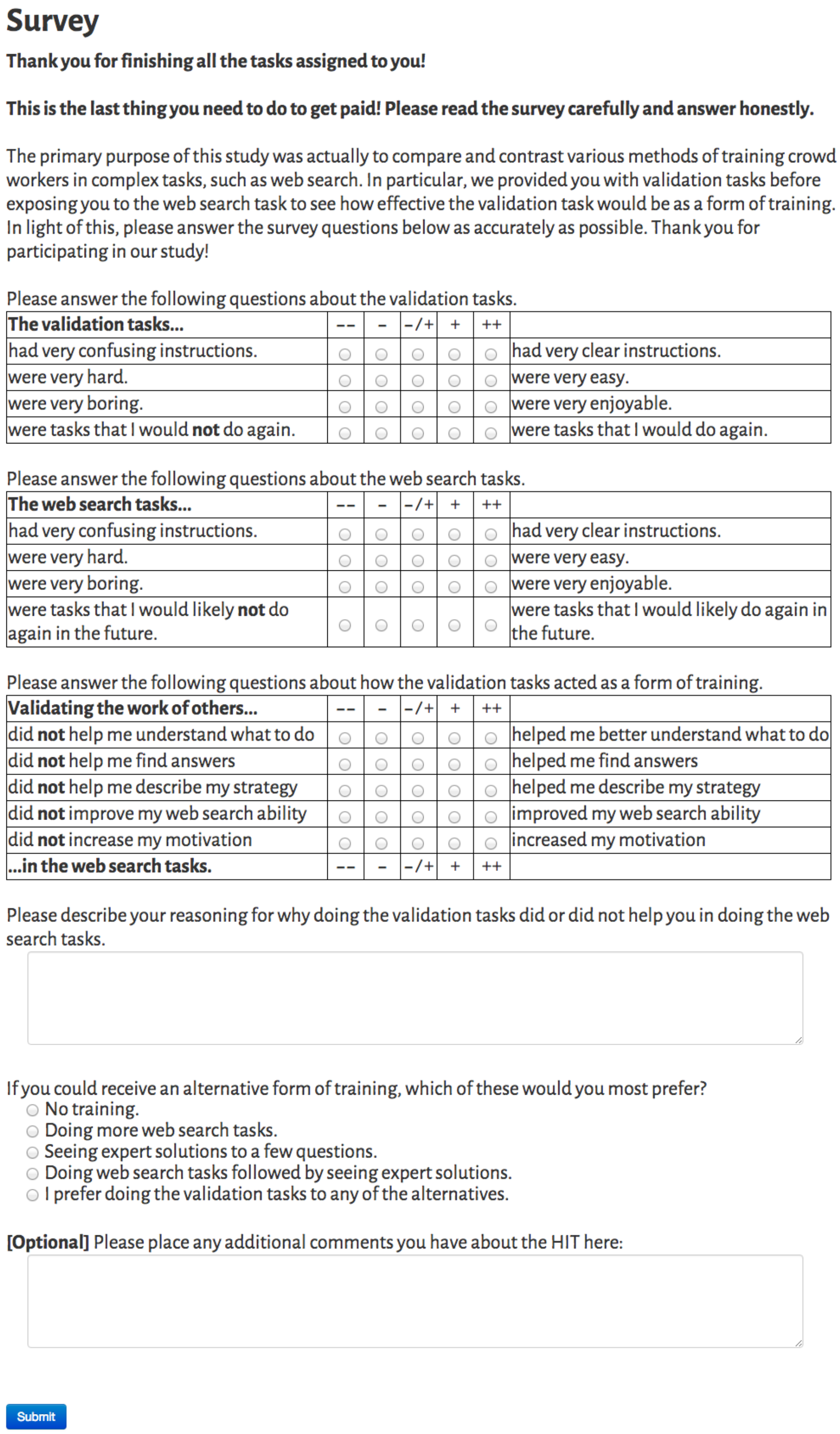
Please describe your reasoning for why reviewing the expert examples did or did not help you in doing the web search tasks.

If you could receive an alternative form of training, which of these would you most prefer?
○ No training.
○ Doing more web search tasks.
○ Doing web search tasks followed by seeing expert solutions.
○ Validating the solutions of other workers.
○ I prefer seeing expert solutions to any of the alternatives.

[Optional] Please place any additional comments you have about the HIT here:

Thanks for your participation! Remember, please do not discuss the content of this task on any forums or with any other Turkers!

Figure A.5: Example condition survey



Survey

Thank you for finishing all the tasks assigned to you!

This is the last thing you need to do to get paid! Please read the survey carefully and answer honestly.

The primary purpose of this study was actually to compare and contrast various methods of training crowd workers in complex tasks, such as web search. In particular, we provided you with validation tasks before exposing you to the web search task to see how effective the validation task would be as a form of training. In light of this, please answer the survey questions below as accurately as possible. Thank you for participating in our study!

Please answer the following questions about the validation tasks.

The validation tasks...	––	–	–/+	+	++	
had very confusing instructions.	○	○	○	○	○	had very clear instructions.
were very hard.	○	○	○	○	○	were very easy.
were very boring.	○	○	○	○	○	were very enjoyable.
were tasks that I would **not** do again.	○	○	○	○	○	were tasks that I would do again.

Please answer the following questions about the web search tasks.

The web search tasks...	––	–	–/+	+	++	
had very confusing instructions.	○	○	○	○	○	had very clear instructions.
were very hard.	○	○	○	○	○	were very easy.
were very boring.	○	○	○	○	○	were very enjoyable.
were tasks that I would likely **not** do again in the future.	○	○	○	○	○	were tasks that I would likely do again in the future.

Please answer the following questions about how the validation tasks acted as a form of training.

Validating the work of others...	––	–	–/+	+	++	
did **not** help me understand what to do	○	○	○	○	○	helped me better understand what to do
did **not** help me find answers	○	○	○	○	○	helped me find answers
did **not** help me describe my strategy	○	○	○	○	○	helped me describe my strategy
did **not** improve my web search ability	○	○	○	○	○	improved my web search ability
did **not** increase my motivation	○	○	○	○	○	increased my motivation
...in the web search tasks.	––	–	–/+	+	++	

Please describe your reasoning for why doing the validation tasks did or did not help you in doing the web search tasks.

If you could receive an alternative form of training, which of these would you most prefer?

○ No training.
○ Doing more web search tasks.
○ Seeing expert solutions to a few questions.
○ Doing web search tasks followed by seeing expert solutions.
○ I prefer doing the validation tasks to any of the alternatives.

[Optional] Please place any additional comments you have about the HIT here:

Figure A.6: Validation condition instructions

Appendix B

Details of Studies in Empirical Review

For the studies that found a significant difference, Table B.1 gives more details about the technical aspects of each study, such as the form of RL (online vs. offline), the models that were used, the nature of the RL-induced and baseline policies, the outcome variables of interest, and the effect sizes. Table B.2 reports the same information for the remainder of the studies. The outcome variables used in the studies include posttest score, learning gains (i.e., posttest - pretest), normalized learning gains (NLG), time (i.e., how long it takes to reach some desired level of completion), performance (i.e., how well the student performs on some tasks *during* the intervention), and time per problem (i.e., how long students spend on the assigned problems on average). We note that many studies report on more than one outcome variables; in such cases we chose to only report one outcome variable, tending to favor the outcome variable closest to what the instructional policies were directly optimizing. The description of the variety of models used in these studies (as well as acronyms used in Tables B.1 and B.2) is given in Appendix B.1; similarly a description of the variety of RL-induced and baseline policies is given in Appendix B.2. Appendix B.3 describes how some studies performed model selection or policy selection (i.e., how they chose which models or policies to use). There is of course a lot of relevant detail for each study that could not be encapsulated in these tables; we refer the reader to the individual papers for a better understanding of the experiments conducted.

B.1 Models of Learning

Several models of learning have been used to derive instructional policies. Some of these models are based on psychological theories, whereas others were devised from a more machine learning or data-driven perspective. Here we summarize the variety of models used in the empirical studies along with the acronyms used for the models in Tables B.1 and B.2. We begin by describing the models used for vocabulary learning and paired-association tasks in the first wave:

- One-Element Model (**OEM**): This is one of the simplest of models belonging to a class of *all-or-none models* that describe learning in all-or-none fashion: you either know something or you don't. The OEM supposes that for each item, the student is either in a latent learned state or unlearned state. If the student is in the learned state, they will always answer questions on the item correctly, but if they are in the unlearned state, they have a probability of guessing, but will otherwise answer incorrectly. With each item presentation, the student also has some probability of learning the item.

- Single-Operator Linear Model (**SOL**): In opposition to the OEM, this model assumes learning occurs incrementally. In particular, with each item presentation, the probability of a student answering incorrectly decreases by a constant multiplier.
- Random-Trial Increment (**RTI**): This model combines OEM and SOL; with each item-presentation there is some probability that the the student will incrementally learn the skill (i.e., that the probability of answering incorrectly decreases by a constant multiplier).
- Three-State Markov Model (**3SM**): This describes a variety of models that consist of three latent knowledge states (instead of two as in OEM). When the student is in the intermediary state, there is some probability that they will forget the item. Forgetting can occur with each presentation of any other item; therefore, unlike all the previous models, a student's knowledge of an item can change even when the student is practicing other items. These models allow for spacing effects. We will differentiate between two cases of this model: one where all items are assumed to be homogeneous or have the same parameters (**3SM-Hom**), and one where the items can be heterogeneous or each item is allowed to have different parameters (**3SM-Het**). Many years after the experiments performed by Atkinson and colleagues, Katsikopoulos et al. [2001] builds on this work by using a four-state Markov model (**4SM**).

Although these models are relatively simple and all assume that items are independent of one another, several interesting properties emerge when considering using these models to derive instructional policies. First, we note that if we consider the problem of which item to present to a student, then all of the models above can be described as factored POMDPs (where the state can be factored into individual items, and there is a separate transition and observation matrix for each item), provided that we give an appropriate reward function, such as a delayed reward proportional to the number of items learned at the end of the study. The SOL model in particular can be described as a factored MDP, since we can determine the state of each item uniquely by how many times it was presented to the student. Moreover, SOL is a deterministic MDP. Thus, the optimal policy for SOL is a non-adaptive policy (or response insensitive strategy, as it is referred to in the literature). In particular, if we assume homogeneous parameters, the optimal policy is to repeatedly cycle through all items in any order to maximize the minimum number of presentations for each item. Even though this is the optimal policy for a model, it is such a simple instructional policy that it is commonly used as a baseline in many of the empirical studies.

On the other hand, for OEM, the optimal strategy is adaptive (or response sensitive). In particular, as shown by Karush and Dear [1967], the optimal policy for OEM (assuming homogeneous parameters) is simply to give the item that has had the shortest consecutive streak of correct responses. Interestingly, as with SOL, (1) the myopic policy is optimal, and (2) this optimal policy does not depend on the parameters of the OEM model, so one does not need to learn the parameters of the model in order to execute its optimal policy. Even though no learning is necessary, we still consider the optimal policy for the OEM model to be an RL-induced policy, albeit a simple one.

We now consider models that have been used in more recent papers:

- Bayesian Knowledge Tracing (**BKT**): This model is identical to OEM, except that it also allows for a probability of slipping when the item or skill is in the learned state. In addition, it is often used in a different context than OEM. In particular, an instructional action (such as a problem in an ITS) may have several different skills on it. When a student works on a problem, their knowledge state of several skills may change. Thus an optimal policy for BKT may be much more complicated than for OEM; however, a heuristic policy is typically used for BKT where problems are given until a student is believed to be in the learned state with high probability (e.g., at least 95%). David et al. [2016] used a modified version of BKT to choose problems believed to be in the student's zone of proximal development.

- Performance Factors Analysis (**PFA**): This is a model that was proposed as an alternative to BKT in the educational data mining literature [Pavlik Jr et al., 2009]. It models the probability of answering and item correctly as a logistic function that depends on the number of times the item was answered correctly in the past and the number of times it was answered incorrectly. Although it is commonly used to predict learning, Papoušek et al. [2016] are the only ones who have evaluated its efficacy in instructional sequencing to our knowledge. Papoušek et al. [2016] use a modified version of the PFA algorithm by combining it with the Elo raing system [Pelánek et al., 2017].

- Featurized (PO)MDP (**Feat-(PO)MDP**): This type of model has been popular in recent papers that consider using RL for instructional sequencing. The idea is to create a MDP (or POMDP) that uses a set of features collected from the software to encapsulate the state of the MDP (or observation space of a POMDP). Features could include, for example, whether the student answered questions of various types correctly in the past, actions taken by the student in an ITS, and aspects of the history of instructional actions such as how many actions of type *X* the policy has given so far. Since there are many features one could feasibly consider, some papers have looked at various methods of doing feature reduction [Chi et al., 2009, Zhou et al., 2017] and feature compression [Mandel et al., 2014b]. For instance, Mandel et al. [2014b] used features to represent the observation space of a POMDP; however, to make the problem tractable, they did feature compression on thousands of features, which resulted in only two features.

- Factored MDP (**FMDP**): This is a particular type of featurized MDP, where the state space is factored into a set of features and each feature's state only depends on some (ideally small) subset of features. This is a useful way to represent how students learn a set of interrelated skills, where the ability to learn one skill might depend on some but not all of the remaining skills. The relationship between skills can be determined by a domain expert (e.g., through a prerequisite graph) [Green et al., 2011] or determined automatically [Doroudi et al., 2017a].

- **POMDP**: All of the models above could be described as a particular type of POMDP, but some papers explicitly use the POMDP formulation to describe their models of learning. Rafferty et al. [2016a] and Whitehill and Movellan [2017] use POMDPs to naturally describe how learners perform concept learning tasks based on cognitive science theories. For example, Rafferty et al. [2016a] use models where the state space consists of hypotheses over concepts, and when the student is presented information that goes against their hypothesis, they randomly transition to a new hypothesis that is in line with the evidence presented.

- **ACT-R**: ACT-R is a cognitive architecture that generally describes human cognition and how people acquire procedural and declarative knowledge [Anderson, 1993]. Pavlik and Anderson [2008] extended ACT-R and used it to derive an instructional policy for sequencing vocabulary items.
- **DASH**: DASH combines a data-driven logistic regression model with psychological theories of memory that captures "difficulty, ability, and student history" [Lindsey et al., 2014b].
- **Machine Learning Models**: Some papers have assumed that the learner learns according to a particular type of machine learning model. For example, Sen et al. [2018] assume the learner learns according to an artificial neural network (**ANN**) and Geana [2015] assumes that learners is either a reinforcement learning agent or a Bayesian learner. Notice that assuming the student is an RL agent is different from assuming that the congitive state of the student changes according to a MDP. Rather, it means that the student is an RL agent that is trying to learn the parameters of an MDP.

Paper(s)	RL Setting	Model	Adaptive Policies	Baseline Policies	Outcome Variable	Effect Size
Laubsch [1969]	Online	1. RTI 2. OEM	1. Myopic-1 2. Optimal	Random Cycle	Posttest	1: d = 1.02 2. Not sig.
Atkinson and Lorton [1969]	Online	OEM	Myopic-1	Random Cycle	Posttest	d = 0.96
Atkinson [1972b]	Offline	1. 3SM-Hom 2. 3SM-Het	Myopic-1	A. Random Cycle B. Student Choice	Posttest Posttest	1vB: 36% increase 2vB: Not sig
Chiang [1974] Exp 1	Online	1. 3SM-Hom 2. 3SM-Het	Myopic-1	Random Cycle	Posttest	1: ? 2: ?
Chiang [1974] Exp 2	Mixed[a]	3SM-Het	3SM-Hom	Random	Response Latency[b]	?
Beck et al. [2000]	Offline	Model-Free	TD(0)-based	Heuristic	Time Per Problem	30% reduction
Katsikopoulos et al. [2001] Exp 1	Offline	4SM-Hom	Myopic-1	Random Cycle	Posttest	d = 1.04
Pavlik and Anderson [2008]	Offline	ACT-R	Myopic-1	A. Cycle-till-Correct B. 3SM-Het	Posttest	B: d = 0.796
Chi et al. [2010a,b]	Offline	Feat-MDP	Optimal	Inverse	Adjusted Posttest	d = 0.86
Lindsey et al. [2014b]	Online	DASH	Threshold	A. Massed B. Spaced	Posttest	B: d = 1.05
Lindsey [2014]	Mixed	DASH	Threshold	A. Random B. Spaced	Posttest	A: d = 0.18[c]
Mandel et al. [2014b]	Offline	Feat-POMDP	1. QMDP Optimal 2. Best Fixed	A. Expert B. Random	# of Levels Completed	1vB: 32% increase 2vB: Not sig
Lin et al. [2015] Exp 1	Online	Model-Free	Genetic Algorithm	Inverse	Posttest	d = 1.29
Lin et al. [2015] Exp 2	Online	Model-Free	Genetic Algorithm	Inverse	Posttest	d = 1.48
Rafferty et al. [2011]	Offline	POMDP	Myopic-2	Random	Time	46% faster
Rafferty et al. [2016a] Exp 1	Offline	POMDP	1. Myopic-2 2. Max Info Gain	Random	Time	1. 54% faster 2. 57% faster
Papoušek et al. [2016]	Offline	PFA + Elo	Threshold[d]	Random	Test Items	?
Zhou et al. [2017]	Offline	Feat-MDP	Optimal	Random	Adjusted Posttest	d = 0.43
Leyzberg et al. [2018]	Online	3SM	Myopic-1	Random	Posttest	d = 2.47
Sen et al. [2018]	Offline	ANN	Fixed	A. Expert B. Random	Posttest	A. d = 0.16 2. d = 0.14

[a] In this experiment, data from Chiang [1974] Exp 1 were used to initialize the model parameters, which was expected to help the 3SM-Het model, since this model has individual parameters for each item. Additionally, a different form of the 3SM model was used in Exp 2, determined by comparing model fits to data from Exp 1.

[b] Chiang [1974] Exp 2 actually used posttest scores as an outcome variable, but no significant difference was found between policies. However, since they found a significant difference in response latency, we included this study among the ones that found a significant difference.

[c] The effect is likely deflated as students found a way to skip review words, which was the content that was being sequenced.

[d] Papoušek et al. [2016] actually test three different adaptive policies: one that adaptively chooses the question and randomly chooses the set of distractors, one that randomly chooses the question and adaptively chooses the distractors, and one that adaptively chooses both. Interestingly, they find that only adaptively choosing the distractors to meet a threshold level of difficulty seems to make a significant difference.

Table B.1: Technical details and effect sizes of the experiments performed by all empirical studies where adaptive policies significantly outperformed all baselines. Descriptions of the models and policies are provided in the main text. Some experiments use multiple adaptive policies, where either the model differs or type of policy differs. In such cases the different models/policies are designated by a number. Similarly, when multiple baseline policies are used, the various baseline policies are designated by a letter. The effect size column shows the effect of each adaptive policy compared to the baseline policy that was found to perform best (e.g., 1vB indicates adaptive policy 1 compared to baseline policy B, where B was the best baseline policy). Cohen's d effect sizes are presented when they were given or could be derived; in other cases the percentage improvement over the baseline is presented.

Paper(s)	RL Setting	Model	Adaptive Policies	Baseline Policies	Outcome Variable	Effect Size
Shen and Chi [2016b]	Offline	Feat-MDP	Optimal	Random	NLG	d = 1.01 (ATI)
Shen et al. [2018a] Exp 1	Offline	POMDP	Optimal?	Random	NLG	d = 0.82 (ATI)
Shen et al. [2018a] Exp 2	Offline	1. POMDP 2. Model-Free	1. Optimal? 2. DQN	Random	1. Time 2. NLG	1. d = 1.01 (ATI) 2. Not sig
Shen et al. [2018b]	Offline	1. Feat-MDP 2. Feat-POMDP	1. ? 2. ?	Random	Posttest Posttest	1: d = 0.83 (ATI) 2. Not sig
Lindsey et al. [2013]	Online	Model-Free	BO	A. Max Fade[a] B. Max Fade + Block C. Intermediate	Posttest	A. 12.6% increase B. Not sig C. Not sig
Rafferty et al. [2016a] Exp 2	Offline	POMDP	1. Myopic-2 2. Max Info Gain	Random	Time	1. Not always sig[b] 2. Sig worse
Whitehill and Movellan [2017]	Offline	POMDP	Policy Gradient	A. Random B. Heuristic	Time	A. 27% faster B. Not sig
Green et al. [2011] Exp 1	Offline	FMDP	Optimal	A. Random B. Heuristic	Posttest	A. $d \approx 3.9$ B. Not sig
Green et al. [2011] Exp 2	Offline	FMDP	Optimal	A. Random B. Expert[c]	Posttest	A. ? B. Not sig
Dear et al. [1967]	Offline	OEM	Optimal	Random Cycle	Posttest	Not sig
Katsikopoulos et al. [2001] Exp 2	Offline	OEM	1. Myopic-1 2. Myopic-1 + Lag	Random Cycle	Posttest	1. Not sig 2. Not sig
Chi et al. [2009]	Offline	Feat-MDP	Optimal	Random	Posttest	Not sig
Rowe [2013]	Offline	Feat-MDP	Optimal	Random	Posttest	Not sig
Clement et al. [2015]	Online	Model-Free	MAB	Inc Difficulty	Difficulty Level Completed	Not sig
Geana [2015]	Offline	RL Agent Bayesian Learner	1. Myopic 2. Myopic	A. Inverse B. Inverse	Posttest	1vA: d = 0.45 1vB: Not sig?[d] 2vB: Not sig
David et al. [2016]	Offline	BKT-based	Threshold	Inc Difficulty	Performance	Not sig
Schatten [2017]	Offline	Matrix Factorization	Threshold	Expert	Posttest	Not sig
Doroudi et al. [2017a]	Offline	1. FMDP 2. FMDP 3. BKT	1. Mastery 2. Myopic-2 3. Mastery	Spiral Difficulty	Posttest	Not sig
Section 8.5.1	Offline	BKT	Inc Time	Spiral Difficulty	Posttest	Not sig
Segal et al. [2018]	Mixed	1. Model-Free 2. BKT	1. MAB 2. Threshold	Inc Difficulty	Posttest	Not sig Not sig
Mettler et al. [2011]	Offline	3SM-Het	Myopic-1	ARTS	Posttest	d = -0.68

[a] Policies are parameterized by the degree of fading and degree of blocking. See Lindsey et al. [2013] for more details. The three baseline policies are maximum fading and no blocking, maximum fading and blocking, and intermediate fading and blocking.

[b] The policies were tested on three different concept learning tasks. The performance of the algorithms vary. In some cases, the POMDP policies are not significantly better than one of the random control conditions.

[c] Green et al. [2011] also compare their FMDP policy to both expert and random baseline policies that give students a choice of three problems to choose from. The overall finding is that regardless of whether students have a choice, the DBN and expert policies outperform random.

[d] The authors do not actually report the significance of this comparison, but it appears to be not significant.

Table B.2: Technical details and effect sizes of the experiments performed by all empirical studies where adaptive policies *did not* significantly outperform baselines. Details are as described in the caption of Table B.1. A policy is labeled as “Optimal?” when we could not verify that it was the optimal policy. For policies with a mixed effect, the effect of comparing the adaptive policy vs. each baseline is shown (rather than just the best baseline).

B.2 Instructional Policies

B.2.1 Model-Based Policies

Given a model, there are many ways to derive an instructional policy. In this section, we will mention some of the most common types of policies used for model-based methods:

- **Optimal**: This refers to the optimal policy given a model. Optimal policies for MDPs can be derived using dynamic programming methods such as value iteration and policy iteration. As mentioned earlier, for simple models such as SOL and OEM, the optimal policy takes a very simple form that does not actually depend on the models. Optimal policies are typically only used for these simple models as well as for feature-based MDPs. For more complex models such as POMDPs and FMDPs with large state spaces, solving for the optimal policy can be intractable. An approximation can be made for POMDPs by solving for the optimal policy in the equivalent QMDP, which is a POMDP where the state is assumed to be known after one action.

- **Myopic**: These policies optimize the reward function by only considering the consequences of the next action (Myopic-1) or the next two actions (Myopic-2), rather than considering the entire horizon. In some simple cases the myopic strategy might be optimal, and in other cases it might be close to optimal Matheson [1964].

- **Threshold**: Given a model that predicts the probability that a student will answer any item correctly, a commonly used policy is to pick the item closest to some threshold. Interestingly, such policies are motivated in terms of educational theories such as desirable difficulty [Lindsey et al., 2014b] and the zone of proximal development [Schatten, 2017]. Khajah et al. [2014] showed in simulation that a threshold policy could be nearly optimal according to two different models of student learning for tasks where there are a set of independent items like paired-association tasks.

- **Fixed**: A fixed policy is a non-adaptive policy that is meant to be near optimal according to a given model. Mandel et al. [2014b] used the best fixed policy according to their model, while Sen et al. [2018] used a hill-climbing approach to find a good fixed policy that led to the least error for their neural network model (although the policy found could have achieved a local optimum).

B.2.2 Model-Free Policies

A minority of the papers that derived RL-induced policies for instructional sequencing did so using model-free methods. In some cases, authors used RL methods such as TD(0) [Beck et al., 2000] and the Deep Q-Network (DQN) [Shen et al., 2018b]. In other cases related methods such as Bayesian optimization (BO), multi-armed bandits (MABs), and genetic algorithms were used. While MABs are typically used to find the best static decision (assuming no change in state), researchers have used them in novel ways to account for student learning, such as by progressing students through a knowledge graph and using the multi-armed bandit to find the best actions among the set that the student is ready for [Clement et al., 2015] or by modifying which actions are optimal by adjusting the weights of actions based on the level of difficulty student can currently tackle [Segal et al., 2018]. Like the model-based threshold policies, these policies also motivate the idea of a set of appropriately challenging problems at any given time using educational theory such as the zone of proximal development [Clement et al., 2015, Segal et al., 2018].

B.2.3 Baseline Policies

RL-induced policies have been compared to a variety of baseline policies. Some of the common baseline policies include:

- **Random**: This policy presents content in a random order. Although random sequencing will often be a weak baseline, it can be reasonable in cases where the content being sequenced does not have strong dependencies, especially given that interleaving content has been shown to be effective.
- **Random Cycle**: This policy randomly cycles through items, such as words in a paired-association task, in a fixed order. As mentioned earlier, this is actually the optimal policy under the simple SOL model. However, because it is one of the simplest policies that could be considered and a non-adaptive policy, we consider it as a baseline policy whenever it is used.
- **Inverse**: This policy takes the reward function of an MDP and minimizes it instead of maximizing it. Therefore, it is a policy that is made to intentionally perform poorly. The idea behind using such a policy is to show that the ordering of instructional activities actually makes a difference. However, this baseline cannot be used to discern if RL-induced policies are effective ways of teaching students beyond reasonable methods for instructional sequencing.
- **Inc Difficulty**: This general type of policy orders content in order of increasing difficulty or complexity as determined by domain experts. In our experiments, we used a modified version of this general policy referred to as **Spiral Difficulty** where three broad topics were ordered in terms of difficulty, but once students finished twelve problems per topic, they would be returned to the first topic and so on, loosely motivated by the idea of a spiral curriculum [Bruner, 1960, Harden, 1999].

B.3 Model/Policy Selection

In Tables B.1 and B.2, we report the models ultimately used in each of the studies; however, many of the researchers considered a variety of models and policies before settling on one in particular. These researchers used a variety of model selection (or policy selection) criteria to choose which policy to use in the experiments. For a number of different studies, Min Chi and colleagues [Chi et al., 2009, 2010b, Shen and Chi, 2016b, Shen et al., 2018a, Zhou et al., 2017] fit several MDPs on different featurized representations of the state space and used the expected cumulative reward (ECR) of the optimal policies under each model to determine which policy was expected to perform best. As shown by Mandel et al. [2014b], ECR is a biased and inconsistent estimator of the true value of a policy. Instead, Mandel et al. [2014b] used importance sampling to evaluate policies for several different types of models to settle on a final policy. While the importance sampling estimator can give an unbiased estimate of the value of a policy, it is only data-efficient when sequencing a few instructional activities and can lead to biased decisions when used for policy selection [Doroudi et al., 2017c].

To provide a more robust estimator than ECR while mitigating the data inefficiency of importance sampling, we proposed the robust evaluation matrix (REM) method to perform policy selection [Doroudi et al., 2017a]. REM involves simulating each instructional policy of interest using multiple plausible models of student learning that were fit to previously collected data. If a policy robustly outperforms other policies according to multiple different models, we can have increased confidence that it will actually be a better instructional policy rather than "over-fitting" to a particular model (which ECR is susceptible to). We used REM to select a policy for our second experiment as described in Section 8.5.1. Several other studies used similar robust evaluation methods [Lindsey et al., 2014b, Rafferty et al., 2016a].

Bibliography

Umair Z Ahmed, Sumit Gulwani, and Amey Karkare. Automatically generating problems and solutions for natural deduction. In *Twenty-Third International Joint Conference on Artificial Intelligence*, pages 1968–1975, 2013.

Turadg Aleahmad, Vincent Aleven, and Robert Kraut. Creating a corpus of targeted learning resources with a web-based open authoring tool. *IEEE Transactions on Learning Technologies*, 2(1):3–9, 2009.

Turadg Aleahmad, Vincent Aleven, and Robert Kraut. Automatic rating of user-generated math solutions. In *Proceedings of the 3rd International Conference on Educational Data Mining*. International Educational Data Mining Society, 2010.

Vincent Aleven, Bruce McLaren, Jonathan Sewall, and Kenneth R Koedinger. Example-tracing tutors: A new paradigm for intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 19(2):105, 2009.

Vincent Aleven, Elizabeth A McLaughlin, R Amos Glenn, and Kenneth R Koedinger. Instruction based on adaptive learning technologies. *Handbook of research on learning and instruction*, 2016a.

Vincent Aleven, Franceska Xhakaj, Kenneth Holstein, and Bruce M McLaren. Developing a teacher dashboard for use with intelligent tutoring systems. In *IWTA@ EC-TEL*, pages 15–23, 2016b.

Russell G Almond. An illustration of the use of markov decision processes to represent student growth (learning). *ETS Research Report Series*, 2007(2), 2007.

Per-Arne Andersen, Christian Kråkevik, Morten Goodwin, and Anis Yazidi. Adaptive task assignment in online learning environments. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, page 5. ACM, 2016.

John R Anderson. *Rules of the Mind*. Lawrence Erlbaum Associates, 1993.

John R Anderson, C Franklin Boyle, and Brian J Reiser. Intelligent tutoring systems. *Science*, 228(4698):456–462, 1985.

Rika Antonova, Joe Runde, Min Hyung Lee, and Emma Brunskill. Automatically learning to teach to the learning objectives. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 317–320. ACM, 2016.

Richard C Atkinson. Ingredients for a theory of instruction. *American Psychologist*, 27(10):921, 1972a.

Richard C Atkinson. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology*, 96(1):124, 1972b.

Richard C Atkinson. Computer assisted instruction: Optimizing the learning process. In *Annual Convention of the Association for Psychological Science*, 2014.

Richard C Atkinson and Robert C Calfee. Mathematical learning theory. Technical Report 50, Institute of Mathematical Studies in the Social Sciences, 1963.

Richard C Atkinson and Paul Lorton, Jr. Computer-based instruction in spelling: An investigation of optimal strategies for presenting instructional material. final report. Technical report, U.S. Department of Health, Education, and Welfare, 1969.

Anne Aula and Daniel M Russell. Complex and exploratory web search. In *Information Seeking Support Systems Workshop (ISSS 2008), Chapel Hill, NC, USA*, 2008.

Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss markov random fields and probabilistic soft logic. *arXiv preprint arXiv:1505.04406*, 2015.

Ryan SJd Baker, Albert T Corbett, Kenneth R Koedinger, Shelley Evenson, Ido Roll, Angela Z Wagner, Meghan Naim, Jay Raspat, Daniel J Baker, and Joseph E Beck. Adapting to when students game an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*, pages 392–401. Springer, 2006.

Ryan SJd Baker, Albert T Corbett, and Vincent Aleven. Improving contextual models of guessing and slipping with a truncated training set. *Human-Computer Interaction Institute*, page 17, 2008.

Ryan SJd Baker, Albert T Corbett, Sujith M Gowda, Angela Z Wagner, Benjamin A MacLaren, Linda R Kauffman, Aaron P Mitchell, and Stephen Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 52–63. Springer, 2010a.

Ryan SJd Baker, Albert T Corbett, Sujith M Gowda, Angela Z Wagner, Benjamin A MacLaren, Linda R Kauffman, Aaron P Mitchell, and Stephen Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 52–63. Springer, 2010b.

Ryan SJd Baker, Sujith M Gowda, Michael Wixon, Jessica Kalka, Angela Z Wagner, Aatish Salvi, Vincent Aleven, Gail W Kusbit, Jaclyn Ocumpaugh, and Lisa Rossi. Towards sensor-free affect detection in cognitive tutor algebra. In *Proceedings of the 5th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2012.

Tiffany Barnes and John Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *International Conference on Intelligent Tutoring Systems*, pages 373–382. Springer, 2008.

JE Beck. Modeling the student with reinforcement learning. In *Machine learning for User Modeling Workshop at the Sixth International Conference on User Modeling*. Citeseer, 1997.

Joseph Beck and Xiaolu Xiong. Limits to accuracy: How well can we do at student modeling? In *Proceedings of the 6th International Conference on Educational Data Mining*. International

Educational Data Mining Society, 2013.

Joseph Beck, Beverly Park Woolf, and Carole R Beal. Advisor: A machine learning architecture for intelligent tutor construction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 552–557. AAAI Press, 2000.

Joseph E Beck and Kai-min Chang. Identifiability: A fundamental problem of student modeling. In *International Conference on User Modeling*, pages 137–146. Springer, 2007.

Joseph E. Beck and Yue Gong. Wheel-spinning: Students who fail to master a skill. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 431–440, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

Abdellah Bennane, T D'Hondt, and B Manderick. An approach of reinforcement learning use in tutoring systems. In *Proceedings of the 1st International Conference on Machine Learning and Applications*, page 993, 2002.

Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322. ACM, 2010.

Stephen B Blessing. A programming by demonstration authoring tool for model-tracing tutors. *International Journal of Artificial Intelligence in Education*, 8:233–261, 1997.

Benjamin S Bloom. Learning for mastery. instruction and curriculum. regional education laboratory for the carolinas and virginia, topical papers and reprints, number 1. *Evaluation comment*, 1(2):n2, 1968.

Paul Boekhout, Tamara Gog, Margje WJ Wiel, Dorien Gerards-Last, and Jacques Geraets. Example-based learning: Effects of model expertise in relation to student expertise. *British Journal of Educational Psychology*, 80(4):557–566, 2010.

Gordon H Bower. Application of a model to paired-associate learning. *Psychometrika*, 26(3): 255–280, 1961.

George EP Box. Robustness in the strategy of scientific model building. In *Robustness in statistics*, pages 201–236. Elsevier, 1979.

Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

Jerome S Bruner. *The process of education*. Harvard University Press, 1960.

Emma Brunskill and Stuart Russell. Partially observable sequential decision making for problem selection in an intelligent tutoring system. In *Proceedings of the 4th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2011.

Rafael A Calvo and Sidney D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on affective computing*, 1(1):18–37, 2010.

Hao Cen. *Generalized learning factors analysis: improving cognitive models with machine learning*. PhD thesis, Carnegie Mellon University, 2009.

Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.

John Champaign and Robin Cohen. A model for content sequencing in intelligent tutoring systems based on the ecological approach and its validation through simulated students. In *FLAIRS Conference*, 2010.

Verne G Chant and Richard C Atkinson. Optimal allocation of instructional effort to interrelated learning strands. *Journal of Mathematical Psychology*, 10(1):1–25, 1973.

Devendra Singh Chaplot, Eunhee Rhim, and Jihie Kim. Personalized adaptive learning using neural networks. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 165–168. ACM, 2016.

Binglin Chen, Matthew West, and Craig Ziles. Towards a model-free estimate of the limits to student modeling accuracy. In *Proceedings of the 11th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2018.

Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. Break it down: A comparison of macro-and microtasks. In *Proceedings of CHI*, 2015.

Michelene TH Chi and Ruth Wylie. The icap framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist*, 49(4):219–243, 2014.

Michelene TH Chi, Miriam Bassok, Matthew W Lewis, Peter Reimann, and Robert Glaser. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182, 1989.

Min Chi, Pamela Jordan, Kurth VanLehn, and Moses Hall. Reinforcement learning based feature selection for developing pedagogically effective tutorial dialogue tactics. In *Proceedings of the 1st International Conference on Educational Data Mining*. International Educational Data Mining Society, 2008.

Min Chi, Pamela W Jordan, Kurt Vanlehn, and Diane J Litman. To elicit or to tell: Does it matter? In *Aied*, pages 197–204, 2009.

Min Chi, Kurt VanLehn, and Diane Litman. Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In *International Conference on Intelligent Tutoring Systems*, pages 224–234. Springer, 2010a.

Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. Inducing effective pedagogical strategies using learning context features. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 147–158. Springer, 2010b.

Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180, 2011.

Alice Chiang. *Instructional algorithms derived from mathematical learning models: An application in computer assisted instruction of paired-associated items*. PhD thesis, City University of New York, 1974.

Kwangsu Cho and Charles MacArthur. Learning by reviewing. *Journal of Educational Psychol-*

ogy, 103(1):73, 2011.

Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining (JEDM)*, 7(2):20–48, 2015.

Benjamin Clement, Pierre-Yves Oudeyer, and Manuel Lopes. A comparison of automatic teaching strategies for heterogeneous student populations. In *Proceedings of the 9th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2016.

Jacob Cohen. Statistical power analysis for the behavioral sciences 2nd edn, 1988.

Gemma Corbalan, Liesbeth Kester, and Jeroen JG Van Merriënboer. Selecting learning tasks: Effects of adaptation and shared control on learning efficiency and task involvement. *Contemporary Educational Psychology*, 33(4):733–756, 2008.

Albert Corbett. Cognitive mastery learning in the act programming tutor. In *Papers from the AAAI Spring Symposium*. AAAI Press, 2000.

Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

Geoff D Cumming and JA Self. Learner models in collaborative intelligent educational systems. *Teaching Knowledge and Intelligent Tutoring. Ablex*, 1991.

Sayamindu Dasgupta, William Hale, Andrés Monroy-Hernández, and Benjamin Mako Hill. Remixing as a pathway to computational thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1438–1449. ACM, 2016.

Lucie Daubigney, Matthieu Geist, and Olivier Pietquin. Model-free pomdp optimisation of tutoring systems with echo-state networks. In *SIGDIAL Conference*, pages 102–106, 2013.

Yossi Ben David, Avi Segal, and Ya'akov Kobi Gal. Sequencing educational content in classrooms using bayesian knowledge tracing. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 354–363. ACM, 2016.

Robert E Dear, Harry F Silberman, Donald P Estavan, and Richard C Atkinson. An optimal strategy for the presentation of paired-associate items. *Systems Research and Behavioral Science*, 12(1):1–13, 1967.

Keith Devlin. Maththink mooc v4 - part 6. `https://mooctalk.org/2013/12/23/maththink-mooc-part-6/`, 2013.

Pedro Domingos, Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla. Unifying logical and statistical ai. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*, pages 2–7. AAAI Press, 2006.

Mira Dontcheva, Robert R Morris, Joel R Brandt, and Elizabeth M Gerber. Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3379–3388. ACM, 2014.

Fabiano A Dorça, Luciano V Lima, Márcia A Fernandes, and Carlos R Lopes. Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis. *Expert Systems with Applications*, 40(6):2092–2101, 2013.

Shayan Doroudi and Emma Brunskill. The misidentified identifiability problem of bayesian knowledge tracing. In *Proceedings of the 10th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2017.

Shayan Doroudi and Emma Brunskill. Fairer but not fair enough: On the equitability of knowledge tracing. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 335–339. ACM, 2019.

Shayan Doroudi, Kenneth Holstein, Vincent Aleven, and Emma Brunskill. Towards understanding how to leverage sense-making, induction and refinement, and fluency to improve robust learning. In *Proceedings of the 8th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2015.

Shayan Doroudi, Ece Kamar, Emma Brunskill, and Eric Horvitz. Toward a learning science for complex crowdsourcing tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2623–2634. ACM, 2016.

Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Robust evaluation matrix: Towards a more principled offline exploration of instructional policies. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 3–12. ACM, 2017a.

Shayan Doroudi, Philip S Thomas, and Emma Brunskill. Importance sampling for fair policy selection. In *Uncertainity in Artificial Intelligence*. Association of Uncertainty in Artificial Intelligence, 2017b.

Shayan Doroudi, Philip S Thomas, and Emma Brunskill. Importance sampling for fair policy selection. In *Uncertainity in Artificial Intelligence*. Association of Uncertainty in Artificial Intelligence, 2017c.

Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Otis Dudley Duncan. *Introduction to structural equation models*. Elsevier, 1975.

Hermann Ebbinghaus. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot, 1885.

Elsa Eiriksdottir and Richard Catrambone. Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors*, 53(6):749–770, 2011.

Mohammad H Falakmasir, Zachary A Pardos, Geoffrey J Gordon, and Peter Brusilovsky. A spectral learning approach to knowledge tracing. In *Proceedings of the 6th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2013.

Nancy Falchikov. Peer feedback marking: Developing peer assessment. *Programmed Learning*, 32(2):175–187, 1995.

Alireza Farasat, Alexander Nikolaev, Suzanne Miller, and Rahul Gopalsamy. Crowdlearning: Towards collaborative problem-posing at scale. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 221–224. ACM, 2017.

Giuseppe Fenza, Francesco Orciuoli, and Demetrios G Sampson. Building adaptive tutoring model using artificial neural networks and reinforcement learning. In *Advanced Learning Technologies (ICALT), 2017 IEEE 17th International Conference on*, pages 460–462. IEEE, 2017.

Bill Ferster. *Teaching machines: learning from the intersection of education and technology*. JHU Press, 2014.

Jeremiah Folsom-Kovarik, Gita Sukthankar, Sae Schatz, and Denise Nicholson. Scalable pomdps for diagnosis and planning in intelligent tutoring systems. In *Papers from the AAAI Fall Symposium*, 2010. URL https://www.aaai.org/ocs/index.php/FSS/FSS10/paper/view/2290/2701.

Jeremiah T Folsom-Kovarik. *Leveraging Help Requests in POMDP Intelligent Tutoring Systems*. PhD thesis, University of Central Florida, 2012.

John P Fry. Interactive relationship between inquisitiveness and student control of instruction. *Journal of Educational Psychology*, 63(5):459, 1972.

Andra Geana. *Information sampling, learning and exploration*. PhD thesis, Princeton University, 2015.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Sarah Gielen, Elien Peeters, Filip Dochy, Patrick Onghena, and Katrien Struyven. Improving the effectiveness of peer feedback for learning. *Learning and instruction*, 20(4):304–315, 2010.

Elena L Glassman, Aaron Lin, Carrie J Cai, and Robert C Miller. Learnersourcing personalized hints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1626–1636. ACM, 2016.

Karan Goel, Christoph Dann, and Emma Brunskill. Sample efficient policy search for optimal stopping domains. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1711–1717. AAAI Press, 2017.

Yue Gong and Joseph E Beck. Towards detecting wheel-spinning: Future failure in mastery learning. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 67–74. ACM, 2015.

José P González-Brenes and Yun Huang. Your model is predictive–but is it useful? theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In *Proceedings of the 8th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2015.

Derek T Green, Thomas J Walsh, Paul R Cohen, and Yu-Han Chang. Learning a skill-teaching curriculum with dynamic bayes nets. In *IAAI*, 2011.

Assaf Hallak, COM François Schnitzler, Timothy Mann, and Shie Mannor. Off-policy model-based learning under unknown factored dynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 711–719, 2015.

John D Hansen and Justin Reich. Democratizing education? examining access and usage patterns in massive open online courses. *Science*, 350(6265):1245–1248, 2015.

Ronald M Harden. What is a spiral curriculum? *Medical teacher*, 21(2):141–143, 1999.

Neil T Heffernan, Korinn S Ostrow, Kim Kelly, Douglas Selent, Eric G Van Inwegen, Xiaolu Xiong, and Joseph Jay Williams. The future of adaptive learning: Does the crowd hold the key? *International Journal of Artificial Intelligence in Education*, 26(2):615–644, 2016.

Michael Heilman. Automatic factual question generation from text. *Language Technologies Institute School of Computer Science Carnegie Mellon University*, 195, 2011.

Pamela J Hinds, Michael Patterson, and Jeffrey Pfeffer. Bothered by abstraction: the effect of expertise on knowledge transfer and subsequent novice performance. *Journal of applied psychology*, 86(6):1232, 2001.

William Hoiles and Mihaela Schaar. Bounded off-policy evaluation with missing data for course recommendation and curriculum design. In *International Conference on Machine Learning*, pages 1596–1604, 2016.

Kenneth Holstein. Towards teacher-ai hybrid systems. In *Companion Proceedings of the Eigth International Conference on Learning Analytics & Knowledge*, 2018.

Kenneth Holstein, Bruce M. McLaren, and Vincent Aleven. Student learning benefits of a mixed-reality teacher awareness tool in ai-enhanced classrooms. In Carolyn Penstein Rosé, Roberto Martínez-Maldonado, H. Ulrich Hoppe, Rose Luckin, Manolis Mavrikis, Kaska Porayska-Pomsta, Bruce McLaren, and Benedict du Boulay, editors, *Artificial Intelligence in Education*, pages 154–168, Cham, 2018. Springer International Publishing.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Ronald A Howard. *Dynamic Programming and Markov Processes*. John Wiley, Oxford, England, 1960a.

Ronald A Howard. Machine-aided learning. *High speed computer system research: quarterly progress report*, 9:19–20, 1960b.

Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

David Hu. How khan academy is using machine learning to assess student mastery, 2011. URL `http://david-hu.com/2011/11/02/how-khan-academy-is-using-machine-learning-to-assess-student-mastery.html`.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*, 2016.

Yi-Ting Huang, Ya-Min Tseng, Yeali S Sun, and Meng Chang Chen. TEDQuiz: Automatic quiz generation for TED talks video clips to assess listening comprehension. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pages 350–354. IEEE, 2014.

Yun Huang, Jose Gonzalez-Brenes, Rohit Kumar, and Peter Brusilovsky. A framework for multifaceted evaluation of student models. In *Proceedings of the 8th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2015.

Anette Hunziker, Yuxin Chen, Oisin Mac Aodha, Manuel Gomez Rodriguez, Andreas Krause, Pietro Perona, Yisong Yue, and Adish Singla. Teaching multiple concepts to forgetful learners. *arXiv preprint arXiv:1805.08322*, 2018.

Ana Iglesias, Paloma Martinez, and Fernando FernÃąndez. An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. *Informatics in Education*, 2:223–240, 10 2003.

Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. Learning pedagogical policies from few training data. In *Proceedings of the 17th European Conference on Artificial Intelligence Workshop on Planning, Learning and Monitoring with Uncertainty and Dynamic Worlds*, 2006.

Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence*, 31(1):89–106, 2009.

Matthew P Jarvis, Goss Nuzzo-Jones, and Neil T Heffernan. Applying machine learning techniques to rule generation in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*, pages 541–553. Springer, 2004.

Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S Srinivasa, and J Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research*, page 0278364918776060, 2018.

Nan Jiang and Lihong Li. Doubly robust off-policy evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Samuel RH Joseph, Andrew Smith Lewis, and Michael H Joseph. Adaptive vocabulary instruction. In *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*, pages 141–145. IEEE, 2004.

Eunice Jun, Morelle Arian, and Katharina Reinecke. The potential for scientific outreach and learning in mechanical turk experiments. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, page 3. ACM, 2018.

Slava Kalyuga and John Sweller. Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. *Educational Technology Research and Development*, 53(3): 83–93, 2005.

Slava Kalyuga, Paul Chandler, Juhani Tuovinen, and John Sweller. When problem solving is superior to studying worked examples. *Journal of educational psychology*, 93(3):579, 2001.

Slava Kalyuga, Paul Ayres, Paul Chandler, and John Sweller. The expertise reversal effect. *Educational psychologist*, 38(1):23–31, 2003.

William Karush and RE Dear. Optimal strategy for item presentation in a learning process.

Management Science, 13(11):773–785, 1967.

Tanja Käser, Severin Klingler, and Markus Gross. When to stop?: towards universal instructional policies. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 289–298. ACM, 2016.

Konstantinos V Katsikopoulos, Donald L Fisher, and Susan A Duffy. Experimental evaluation of policies for sequencing the presentation of associations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(1):55–59, 2001.

Kim Kelly, Yan Wang, Tamisha Thompson, and Neil Heffernan. Defining mastery: Knowledge tracing versus n-consecutive correct responses. *STUDENT MODELING FROM DIFFERENT ASPECTS*, page 39, 2016.

Mohammad M Khajah, Robert V Lindsey, and Michael C Mozer. Maximizing students' retention via spaced review: Practical guidance from computational models of memory. *Topics in cognitive science*, 6(1):157–169, 2014.

Juho Kim et al. *Learnersourcing: improving learning with collective learner activity*. PhD thesis, Massachusetts Institute of Technology, 2015.

Mable B Kinzie and Howard J Sullivan. Continuing motivation, learner control, and cai. *Educational Technology Research and Development*, 37(2):5–14, 1989.

Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52. ACM, 2011.

Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013a.

Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013b.

Kenneth R Koedinger and Vincent Aleven. An interview reflection on "Intelligent Tutoring Goes to School in the Big City". *International Journal of Artificial Intelligence in Education*, 26(1): 13–24, 2016.

Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012a.

Kenneth R Koedinger, Elizabeth A McLaughlin, and John C Stamper. Automated student model improvement. In *Proceedings of the 5th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2012b.

Kenneth R Koedinger, Emma Brunskill, Ryan SJd Baker, Elizabeth A McLaughlin, and John Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

Markus Krause, Margeret Hall, Joseph Jay Williams, Praveen Paritosh, John Prip, and Simon Caton. Connecting online work and online education at scale. In *Proceedings of the 2016 CHI*

Conference Extended Abstracts on Human Factors in Computing Systems, pages 3536–3541. ACM, 2016.

Janne V Kujala, Ulla Richardson, and Heikki Lyytinen. A bayesian-optimal principle for learner-friendly adaptation in learning games. *Journal of Mathematical Psychology*, 54(2):247–255, 2010.

Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R Klemmer. Peer and self assessment in massive online classes. In *Design thinking research*, pages 131–168. Springer, 2015.

Suna Kyun, Slava Kalyuga, and John Sweller. The effect of worked examples when learning to write essays in english literature. *The Journal of Experimental Education*, 81(3):385–408, 2013.

Andreas Lachner and Matthias Nückles. Bothered by abstractness or engaged by cohesion? expertsâĂŹ explanations enhance novicesâĂŹ deep-learning. *Journal of Experimental Psychology: Applied*, 21(1):101, 2015.

Aazim Lakhani. Adaptive teaching: learning to teach. Master's thesis, University of Victoria, 2018.

Andrew S Lan and Richard G Baraniuk. A contextual bandits framework for personalized learning action selection. In *Proceedings of the 9th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2016.

Joachim Helmut Laubsch. *An adaptive teaching system for optimal item allocation*. PhD thesis, Stanford University, 1969.

Nguyen-Thinh Le, Tomoko Kojiri, and Niels Pinkwart. Automatic question generation for educational applications–the state of art. In *Advanced Computational Methods for Knowledge Engineering*, pages 325–338. Springer, 2014.

Jung In Lee and Emma Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2012.

Roberto S Legaspi and Raymund C Sison. A machine learning framework for an expert tutor construction. In *Computers in Education, 2002. Proceedings. International Conference on*, pages 670–674. IEEE, 2002.

Daniel Leyzberg, Samuel Spaulding, and Brian Scassellati. Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 423–430. ACM, 2014.

Daniel Leyzberg, Aditi Ramachandran, and Brian Scassellati. The effect of personalization in longer-term robot tutoring. *ACM Transactions on Human-Robot Interaction (THRI)*, 7(3):19, 2018.

Chen Lin and Min Chi. Intervention-bkt: incorporating instructional interventions into bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems*, pages 208–218. Springer, 2016.

Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

Hsuan-Ta Lin, Po-Ming Lee, and Tzu-Chien Hsiao. Online pedagogical tutorial tactics optimization using genetic-based reinforcement learning. *The Scientific World Journal*, 2015.

Robert Lindsey. *Probabilistic Models of Student Learning and Forgetting*. PhD thesis, University of Colorado at Boulder, 2014.

Robert V Lindsey and Michael C Mozer. Predicting and improving memory retention: Psychological theory matters in the big data era. In *Big data in cognitive science*, pages 43–73. Psychology Press, 2016.

Robert V Lindsey, Michael C Mozer, William J Huggins, and Harold Pashler. Optimizing instructional policies. In *Advances in Neural Information Processing Systems*, pages 2778–2786, 2013.

Robert V Lindsey, Mohammad Khajah, and Michael C Mozer. Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in Neural Information Processing Systems*, pages 1386–1394, 2014a.

Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological Science*, 25(3): 639–647, 2014b.

Chung Laung Liu. *A study in machine-aided learning*. PhD thesis, Massachusetts Institute of Technology, 1960.

Derek Lomas, John Stamper, Ryan Muller, Kishan Patel, and Kenneth R Koedinger. The effects of adaptive sequencing algorithms on player engagement within an online game. In *International Conference on Intelligent Tutoring Systems*, pages 588–590. Springer, 2012.

Yanjin Long and Vincent Aleven. Mastery-oriented shared student/system control over problem selection in a linear equation tutor. In *International Conference on Intelligent Tutoring Systems*, pages 90–100. Springer, 2016.

AA Lumsdaine. Teaching machines and self-instructional materials. *Audiovisual Communication Review*, 7(3):163–181, 1959.

Kristi Lundstrom and Wendy Baker. To give is better than to receive: The benefits of peer review to the reviewer's own writing. *Journal of second language writing*, 18(1):30–43, 2009.

Christopher J MacLellan, Erik Harpstead, Rony Patel, and Kenneth R Koedinger. The apprentice learner architecture: Closing the loop between learning theory and educational data. In *Proceedings of the 9th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2016.

Ankit Malpani, Balaraman Ravindran, and Hema Murthy. Personalized intelligent tutoring system using reinforcement learning. In *FLAIRS Conference*, 2011.

Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1077–1084. International Foundation for Autonomous Agents and Multiagent Systems, 2014a.

Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1077–1084. International Foundation for Autonomous Agents and Multiagent Systems, 2014b.

Kimberly N Martin and Ivon Arroyo. Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In *Intelligent Tutoring Systems*, pages 564–572. Springer, 2004.

James Ernest Matheson. *Optimum teaching procedures derived from mathematical learning models*. PhD thesis, Stanford University, 1964.

Noboru Matsuda, William W Cohen, and Kenneth R Koedinger. Teaching the teacher: tutoring SimStudent leads to more effective cognitive tutor authoring. *International Journal of Artificial Intelligence in Education*, 25(1):1–34, 2015.

Noboru Matsuda, Sanjay Chandrasekaran, and John C Stamper. How quickly can wheel spinning be detected? In *Proceedings of the 9th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2016.

Claudia Mazziotti, Wayne Holmes, Michael Wiedmann, Katharina Loibl, Nikol Rummel, Manolis Mavrikis, Alice Hansen, and Beate Grawemeyer. Robust student knowledge: Adapting to individual student needs as they explore the concepts and practice the procedures of fractions. In *Workshop on Intelligent Support in Exploratory and Open-Ended Learning Environments Learning Analytics for Project Based and Experiential Learning Scenarios at the 17th International Conference on Artificial Intelligence in Education (AIED 2015)*, pages 32–40, 2015.

Manuel Mejía-Lavalle, Hermilo Victorio, Alicia Martínez, Grigori Sidorov, Enrique Sucar, and Obdulia Pichardo-Lagunas. Toward optimal pedagogical action patterns by means of partially observable markov decision process. In *Mexican International Conference on Artificial Intelligence*, pages 473–480. Springer, 2016.

Everett Mettler, Christine M Massey, and Philip J Kellman. Improving adaptive learning technology through the use of response times. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*. Cognitive Science Society, 2011.

Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 1969.

Marvin Minsky and Seymour A Papert. Progress report on artificial intelligence. Technical Report AIM-252, Massachusetts Institute of Technology, 1971.

Marvin L Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine*, 12(2):34–34, 1991.

Christopher M. Mitchell, Kristy Elizabeth Boyer, and James C. Lester. A markov decision process model of tutorial intervention in task-oriented dialogue. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 828–831, Berlin, Heidelberg, 2013a. Springer Berlin Heidelberg.

Christopher Michael Mitchell, Kristy Elizabeth Boyer, and James C Lester. Evaluating state representations for reinforcement learning of turn-taking policies in tutorial dialogue. In *SIGDIAL*

Conference, pages 339–343, 2013b.

Tanushree Mitra, CJ Hutto, and Eric Gilbert. Comparing person-and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1345–1354. ACM, 2015.

Piotr Mitros. Learnersourcing of complex assessments. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 317–320. ACM, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Inge Molenaar, Anne Horvers, and Ryan S Baker. Towards hybrid human-system regulation: Understanding children'srl support needs in blended classrooms. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 471–480. ACM, 2019.

Pedro Mota, Francisco Melo, and Luísa Coheur. Modeling students self-studies behaviors. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1521–1528. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

Tong Mu, Shuhan Wang, Erik Andersen, and Emma Brunskill. Combining adaptivity with progression ordering for intelligent tutoring systems. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, page 15. ACM, 2018.

Eni Mustafaraj, Khonzoda Umarova, Franklyn Turbak, and Sohie Lee. Task-specific language modeling for selecting peer-written explanations. In *The Thirty-First International Flairs Conference*, 2018.

Amir Shareghi Najar, Antonija Mitrovic, and Bruce M McLaren. Learning with intelligent tutors and worked examples: selecting learning activities adaptively leads to better learning outcomes than a fixed curriculum. *User Modeling and User-Adapted Interaction*, 26(5):459–491, 2016.

Thomas O Nelson, John Dunlosky, Aurora Graf, and Louis Narens. Utilization of metacognitive judgments in the allocation of study during multitrial learning. *Psychological Science*, 5(4): 207–213, 1994.

Allen Newell. Heuristic programming: Ill-structured problems. *Progress in Operations Research III*, pages 360–414, 1969.

Fleurie Nievelstein, Tamara Van Gog, Gijs Van Dijck, and Henny PA Boshuizen. The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemporary Educational Psychology*, 38(2):118–125, 2013.

MENNO Nijboer. Optimal fact learning: Applying presentation scheduling to realistic conditions. Master's thesis, University of Groningen, 2011.

David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 11(11), 2011.

Seymour Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.

Seymour Papert. Microworlds: transforming education. In *Artificial intelligence and education*, volume 1, pages 79–94. Ablex Norwood, NJ, 1987.

Jan Papoušek, Vít Stanislav, and Radek Pelánek. Evaluation of an adaptive practice system for learning geography facts. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 134–142. ACM, 2016.

Zachary A Pardos and Neil T Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

Philip Pavlik, Thomas Bolster, Sue-Mei Wu, Ken Koedinger, and Brian Macwhinney. Using optimally selected drill practice to train basic facts. In *International Conference on Intelligent Tutoring Systems*, pages 593–602. Springer, 2008.

Philip I Pavlik and John R Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

Radek Pelánek, Jan Papoušek, Jiří Řihák, Vít Stanislav, and Juraj Nižnan. Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction*, 27 (1):89–118, 2017.

Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*, 2013.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015a.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015b.

Olivier Pietquin, Lucie Daubigney, and Matthieu Geist. Optimization of a tutoring system from a fixed set of data. In *SLaTE 2011*, pages 1–4, 2011.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

Anna N. Rafferty, Emma Brunskill, Thomas L. Griffiths, and Patrick Shafto. Faster teaching by pomdp planning. In Gautam Biswas, Susan Bull, Judy Kay, and Antonija Mitrovic, editors, *Artificial Intelligence in Education*, pages 280–287, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via POMDP planning. In *Cognitive Science*, pages 280–287. Springer, 2015a.

Anna N Rafferty, Michelle M LaMar, and Thomas L Griffiths. Inferring learners' knowledge from their actions. *Cognitive Science*, 39(3):584–618, 2015b.

Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via pomdp planning. *Cognitive Science*, 40(6):1290–1332, 2016a.

Anna N Rafferty, Rachel Jansen, and Thomas L Griffiths. Using inverse planning for personalized feedback. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 472–477. International Educational Data Mining Society, 2016b.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

Aditi Ramachandran and Brian Scassellati. Adapting difficulty levels in personalized robot-child tutoring interactions. In *Papers from the 2014 AAAI Workshop*. AAAI Press, 2014.

Martina A. Rau, Vincent Aleven, and Nikol Rummel. Complementary effects of sense-making and fluency-building support for connection making: A matter of sequence? In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 329–338, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Siddharth Reddy, Sergey Levine, and Anca Dragan. Accelerating human learning with deep reinforcement learning. In *NIPS Workshop: Teaching Machines, Robots, and Humans*, 2017.

Siddharth Reddy, Sergey Levine, and Anca Dragan. Shared autonomy via deep reinforcement learning. *arXiv preprint arXiv:1802.01744*, 2018.

Justin Reich and Mizuko Ito. *From good intentions to real outcomes: Equity by design in learning technologies*. Digital Media and Learning Research Hub, 2017.

Alexander Renkl, Robert K Atkinson, and Uwe H Maier. From studying examples to solving problem: Fading worked-out solution steps helps learning. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*. Cognitive Science Society, 2000.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

Frank Restle. The selection of strategies in cue learning. *Psychological Review*, 69(4):329, 1962.

Steven Ritter, John R Anderson, Kenneth R Koedinger, and Albert Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255, 2007.

Joseph Rollinson and Emma Brunskill. From predictive models to instructional policies. In *Proceedings of the 8th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2015.

Yigal Rosen, Ilia Rushkin, Rob Rubin, Liberty Munson, Andrew Ang, Gregory Weber, Glenn Lopez, and Dustin Tingley. The effects of adaptive learning in a massive open online course on learners' skill development. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, page 6. ACM, 2018.

Jonathan P. Rowe and James C. Lester. Improving student problem solving in narrative-centered learning environments: a modular reinforcement learning framework. In Cristina Conati, Neil Heffernan, Antonija Mitrovic, and M. Felisa Verdejo, editors, *Artificial Intelligence in Education*, pages 419–428, Cham, 2015. Springer International Publishing.

Jonathan P Rowe, Bradford W Mott, and James C Lester. Optimizing player experience in interactive narrative planning: A modular reinforcement learning approach. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-14)*. AAAI Press, 2014.

Jonathan Paul Rowe. *Narrative-centered tutorial planning with concurrent Markov decision processes*. PhD thesis, North Carolina State University, 2013.

Philip M Sadler and Eddie Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.

Ron JCM Salden, Vincent Aleven, Rolf Schwonke, and Alexander Renkl. The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science*, 38(3):289–307, 2010a.

Ron JCM Salden, Kenneth R Koedinger, Alexander Renkl, Vincent Aleven, and Bruce M McLaren. Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review*, 22(4):379–392, 2010b.

BH Sreenivasa Sarma and Balaraman Ravindran. Intelligent tutoring systems using reinforcement learning to teach autistic students. In *Home Informatics and Telematics: ICT for The Next Billion*, pages 65–78. Springer, 2007.

Robert Sawyer, Jonathan Rowe, and James Lester. Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning. In Elisabeth André, Ryan Baker, Xiangen Hu, Ma. Mercedes T. Rodrigo, and Benedict du Boulay, editors, *Artificial Intelligence in Education*, pages 323–334, Cham, 2017. Springer International Publishing.

Carlotta Schatten. *Sequencing in Intelligent Tutoring Systems based on online learning Recommenders*. PhD thesis, University of Hildesheim, Germany, 2017.

Carlotta Schatten, Ruth Janning, and Lars Schmidt-Thieme. Vygotsky based sequencing without domain information: a matrix factorization approach. In *International Conference on Computer Supported Education*, pages 35–51. Springer, 2014.

Avi Segal, Yossi Ben David, Joseph Jay Williams, Kobi Gal, and Yaar Shalom. Combining difficulty ranking with multi-armed bandits to sequence educational content. *arXiv preprint arXiv:1804.05212*, 2018.

Ayon Sen, Purav Patel, Martina A Rau, Blake Mason, Robert Nowak, Timothy T Rogers, and Xiaojin Zhu. Machine beats human at sequencing visuals for perceptual-fluency practice. In *Proceedings of the 11th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2018.

Florian Sense. *Making the Most of Human Memory: Studies on Personalized Fact-learning and Visual Working Memory*. PhD thesis, University of Groningen, 2017.

Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1848–1858, 2016.

Shitian Shen and Min Chi. Aim low: Correlation-based feature selection for model-based reinforcement learning. In *Proceedings of the 9th International Conference on Educational Data*

Mining. International Educational Data Mining Society, 2016a.

Shitian Shen and Min Chi. Reinforcement learning: the sooner the better, or the later the better? In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 37–44. ACM, 2016b.

Shitian Shen, Markel Sanz Ausin, Behrooz Mostafavi, and Min Chi. Improving learning & reducing time: A constrained action-based reinforcement learning approach. In *Proceedings of the 2018 Conference on User Modeling Adaptation and Personalization*. ACM, 2018a.

Shitian Shen, Behrooz Mostafavi, Collin Lynch, Tiffany Barnes, and Min Chi. Empirically evaluating the effectiveness of pomdp vs. mdp towards the pedagogical strategies induction. In Carolyn Penstein Rosé, Roberto Martínez-Maldonado, H. Ulrich Hoppe, Rose Luckin, Manolis Mavrikis, Kaska Porayska-Pomsta, Bruce McLaren, and Benedict du Boulay, editors, *Artificial Intelligence in Education*, pages 327–331, Cham, 2018b. Springer International Publishing.

Patrick E Shrout and Joseph L Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420, 1979.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354, 2017.

Rohit Singh, Sumit Gulwani, and Sriram Rajamani. Automatically generating algebra problems. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1620–1627. AAAI Press, 2012.

Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, pages 154–162, 2014.

Richard D Smallwood. *A decision structure for teaching machines*. MIT Press, 1962.

Richard D Smallwood. Optimum policy regions for computer-directed teaching systems. Technical report, U.S. Department of Health, Education, and Welfare, 1968.

Richard D Smallwood. The analysis of economic teaching strategies for a simple learning model. *Journal of Mathematical Psychology*, 8(2):285–301, 1971.

Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

Edward J Sondik. The optimal control of partially observable markov decision processes. *PhD the sis, Stanford University*, 1971.

Samuel Spaulding and Cynthia Breazeal. Learning behavior policies for interactive educational play. In *Models, Algorithms, and HRI Workshop*, 2017.

Rand J Spiro and Michael DeSchryver. Constructivism: When it's the wrong idea and when it's the only idea. In *Constructivist Instruction*, pages 118–136. Routledge, 2009.

Rand J Spiro, Richard L Coulson, Paul J Feltovich, and Daniel K Anderson. Cognitive flexibility theory: Advanced knowledge acquisition in ill-structured domains. Technical Report 441, Center for the Study of Reading, University of Illinois at Urbana-Champaign, 1988.

John C. Stamper and Kenneth R. Koedinger. Human-machine student model discovery and improvement using datashop. In Gautam Biswas, Susan Bull, Judy Kay, and Antonija Mitrovic, editors, *Artificial Intelligence in Education*, pages 353–360, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Monika Steuer, Markus Krause, Maggie Hall, and Steven Dow. On-the-job learning for microtask workers. In *Human Computation 2017 Works-in-Progress*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.

Ryo Suzuki, Niloufar Salehi, Michelle S Lam, Juan C Marroquin, and Michael S Bernstein. Atelier: Repurposing expert crowdsourcing tasks as micro-internships. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2645–2656. ACM, 2016.

John Sweller and Graham A Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):59–89, 1985.

Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schoelkopf, and Manuel Gomez-Rodriguez. Optimizing human learning. *arXiv preprint arXiv:1712.01856*, 2017.

Joshua B Tenenbaum. Rules and similarity in concept learning. In *Advances in neural information processing systems*, pages 59–65, 2000.

Andrew Thatcher. Web search strategies: The influence of web experience and task type. *Information Processing & Management*, 44(3):1308–1329, 2008.

Georgios Theocharous, Richard Beckwith, Nicholas Butko, and Matthai Philipose. Tractable pomdp planning algorithms for optimal teaching in âĂIJspaisâĂİ. In *IJCAI PAIR Workshop*, 2009.

Georgios Theocharous, Nicholas Butko, and Matthai Philipose. Designing a mathematical manipulatives tutoring system using pomdps. In *Proceedings of the POMDP Practitioners Workshop on Solving Real-world POMDP Problems at the 20th International Conference on Automated Planning and Scheduling*, pages 12–16. Citeseer, 2010.

Philip S Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1604.00923*, 2016.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3000–3006. AAAI Press, 2015.

Utkarsh Upadhyay, Abir De, and Manuel Gomez-Rodriguez. Deep reinforcement learning of marked temporal point processes. *arXiv preprint arXiv:1805.09360*, 2018.

Brett van De Sande. Properties of the Bayesian knowledge tracing model. *Journal of Educational Data Mining (JEDM)*, 5(2):1–10, 2013.

Hedderik Van Rijn, Leendert van Maanen, and Marnix van Woudenberg. Passing the test: Improving learning gains by balancing spacing and testing effects. In *Proceedings of the 9th International Conference of Cognitive Modeling*, pages 110–115, 2009.

Kurt VanLehn. Cognitive skill acquisition. *Annual Review of Psychology*, 47(1):513–539, 1996.

Kurt Vanlehn. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.

Kurt VanLehn. Regulative loops, step loops and task loops. *International Journal of Artificial Intelligence in Education*, 26(1):107–112, 2016.

Fangju Wang. Learning teaching in teaching: Online reinforcement learning for intelligent tutoring. In *Future Information Technology*, pages 191–196. Springer, 2014.

Pengcheng Wang, Jonathan Rowe, Bradford Mott, and James Lester. Decomposing drama management in educational interactive narrative: A modular reinforcement learning approach. In *Interactive Storytelling: 9th International Conference on Interactive Digital Storytelling, ICIDS 2016, Los Angeles, CA, USA, November 15–18, 2016, Proceedings 9*, pages 270–282. Springer, 2016.

Pengcheng Wang, Jonathan Rowe, Wookhee Min, Bradford Mott, and James Lester. Interactive narrative personalization with deep reinforcement learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017a.

Pengcheng Wang, Jonathan Rowe, Wookhee Min, Bradford Mott, and James Lester. Simulating player behavior for data-driven interactive narrative personalization. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-17)*. AAAI Press, 2017b.

Lloyd R Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13, 2003.

Daniel S Weld, Eytan Adar, Lydia Chilton, Raphael Hoffmann, Eric Horvitz, Mitchell Koch, James Landay, Christopher H Lin, and Mausam Mausam. Personalized online educationâĂŤa crowdsourcing challenge. In *Papers from the 2012 AAAI Workshop*. AAAI Press, 2012.

Jacob Whitehill and Javier Movellan. Approximately optimal teaching of approximately optimal learners. *IEEE Transactions on Learning Technologies*, 2017.

Jacob Whitehill and Margo Seltzer. A crowdsourcing approach to collecting tutorial videos–toward personalized learning-at-scale. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 157–160. ACM, 2017.

Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Strategies for crowdsourcing social data analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 227–236. ACM, 2012.

Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 379–388. ACM, 2016.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

R Wooley, C Was, Christian D Schunn, and D Dalton. The effects of feedback elaboration on the giver of feedback. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*. Cognitive Science Society, 2008.

Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. Individualized "bayesian knowledge tracing" models. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Ahmed H Zaidi, Russell Moore, and Ted Briscoe. Curriculum q-learning for visual vocabulary acquisition. In *NIPS Workshop: Visually Grounded Interaction and Language*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 217–226. ACM, 2012.

Guojing Zhou, Jianxun Wang, Collin F Lynch, and Min Chi. Towards closing the loop: Bridging machine-induced pedagogical policies to learning theories. In *Proceedings of the 10th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2017.

Li Zhou and Emma Brunskill. Latent contextual bandits and their application to personalized recommendations for new users. *arXiv preprint arXiv:1604.06743*, 2016.

Haiyi Zhu, Steven P Dow, Robert E Kraut, and Aniket Kittur. Reviewing versus doing: Learning and performance in crowd assessment. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1445–1455. ACM, 2014.